Alkalmazott matematikai lapok

1976/1-2

A MAGYAR TUDOMÁNYOS AKADÉMIA
MATEMATIKAI ÉS FIZIKAI TUDOMÁNYOK
OSZTÁLYÁNAK KÖZLEMÉNYEI

2.
KÖTET

AKADÉMIAI KIADÓ, BUDAPEST

A MAGYAR TUDOMÁNYOS AKADÉMIA

MATEMATIKAI ÉS FIZIKAI TUDOMÁNYOK OSZTÁLYÁNAK

ALKALMAZOTT MATEMATIKAI LAPJA

II. kötet 1—2. szám

Szerkesztőség: 1502 Budapest XI., Kende u. 13—17.

Kiadóhivatal: 1055 Budapest V., Alkotmány u. 21.

Az Alkalmazott Matematikai Lapok változó terjedelmű füzetekben jelenik meg, és olyan eredeti tudományos cikkeket publikál, amelyek a gyakorlatban, vagy más tudományokban közvetlenül felhasználható új matematikai eredményt tartalmaznak, illetve már ismert, de színvonalas matematikai apparátus újszerű és jelentős alkalmazását mutatják be. A folyóirat közöl cikk formájában megírt, új tudományos eredménynek számító programokat, és olyan, külföldi folyóiratban már publikált dolgozatokat, amelyek magyar nyelven történő megjelentetése elősegítheti az elért eredmények minél előbbi, széles körű hazai felhasználását.

A folyóirat feladata a Magyar Tudományos Akadémia III. (Matematikai és Fizikai) Osztályának munkájára vonatkozó közlemények, könyvismertetések stb. publikálása is.

Kéziratok a következő címre küldendők:

Prékopa András, felelős szerkesztő
1502 Budapest XI., Kende u. 13—17.

Ugyanerre a címre küldendő minden szerkesztőségi levelezés.

Közlésre el nem fogadott kéziratokat a szerkesztőség lehetőleg visszajuttat a szerzőhöz, de a beküldött kéziratok megőrzéséért vagy továbbításáért felelősséget nem vállal.

Az Alkalmazott Matematikai Lapok előfizetési ára kötetenként 60 forint. Belföldi megrendelések az Akadémiai Kiadó, 1055 Budapest V., Alkotmány u. 21. címen (pénzforgalmi jelzőszám 215—11 488), külföldi megrendelések a Kultúra Külkereskedelmi Vállalat, H-1389 Budapest, Pf. 149. címen (pénzforgalmi jelzőszám 218—10 990) lehetségesek.

A Magyar Tudományos Akadémia III. (Matematikai és Fizikai) Osztálya a következő idegen nyelvű folyóiratokat adja ki:

1. Acta Mathematica Hungaricae,
2. Acta Physica Hungaricae,
3. Studia Scientiarum Mathematicarum Hungarica.

Alkalmazott Matematikai Lapok **2** *(1976) 1—16*

SZTOCHASZTIKUS PROGRAMOZÁSON ALAPULÓ MEGBÍZHATÓSÁGI JELLEGŰ KÉSZLETMODELLEK

PRÉKOPA ANDRÁS és KELLE PÉTER
Budapest

A dolgozatban tárgyalt modellek a PRÉKOPA [7], [12] és ZIERMANN [16] által kidolgozott készletmodellek általánosításai arra az esetre vonatkozólag, amikor egy készletezendő anyag helyett több anyagfajta induló készletmennyiségei felől kell döntenünk, továbbá az egyes anyagok véletlen érkezési folyamatai időben nem feltétlenül homogén sztochasztikus folyamatok. Az érkezési folyamatokról feltesszük, hogy azok egymástól sztochasztikusan függetlenek. A tárgyalandó modellek közül az első megfogalmazása egy korábbi dolgozatban [11] megtalálható. Az ismertetendő modellek sztochasztikus programozási feladatok, az induló készleteket tehát jelen esetben nem formulákkal, hanem algoritmikusan határozzuk meg. Ez mindegyik modell esetében egy-egy nemlineáris programozási feladat megoldásában áll, miközben a valószínűségi feltételben szereplő korlátozó függvény értékeit és gradiensének értékeit szimulációs technikával határozzuk meg. A módszer illusztrálására egy számpéldát is közlünk.

1. Bevezetés

Az ebben a dolgozatban tárgyalt készletezési modellek a PRÉKOPA [7], [12] és ZIERMANN [16] által bevezetett készletmodellek általánosításai. Esetünkben egy készletezendő anyag helyett több anyagfajtáról lesz szó, és nem kívánjuk meg az anyagok beérkezési sztochasztikus folyamatának időbeli homogenitását. Ebben áll az általánosítás.

Ami az induló készletek meghatározását illeti, ez a *Prékopa—Ziermann modellek*ben egyszerű formulák alapján történik. A most ismertetendő modellek esetében viszont a készletszintek meghatározása egy sztochasztikus programozási, egyben nemlineáris programozási feladat megoldása révén történik, melyben bizonyos, valószínűséget jelentő függvény értékeit és gradiensének értékeit szimulációs úton határozzuk meg. Az alkalmazott megoldási módszer tehát bonyolultabb, a modellek viszont némely esetekben közelebb állnak a valósághoz. Többek között azokról az esetekről van szó, amelyekben a szállítási folyamat a vizsgált időszakban koncentrálódik az időszak közepére, vagy végére.

A [7] dolgozatban megfogalmazott legáltalánosabb modell a következő. Jelölje M az induló készletet, T a vizsgált időtartamot, legyen ez mondjuk a $(0, T)$ időintervallum, α_t a t időpontig beérkezett anyagmennyiséget és β_t a szóban forgó anyag iránti, a t időpontig jelentkező felhasználási igényt; meghatározandó az a legkisebb M érték, amelyre

$$P\left(\inf_{0 \leqq t \leqq T} (M+\alpha_t-\beta_t) > 0\right) \geqq 1+\varepsilon, \tag{1.1}$$

ahol ε előre megadott, a gyakorlatban kis számérték, mondjuk pl. $\varepsilon=0{,}05$. Az α_t, β_t sztochasztikus folyamatokra tett feltevések mellett a minimális M esetében

az (1.1) reláció egyenlőséggel teljesül. A kapott egyenlet a megbízhatósági egyenlet nevet viseli.

Az általánosítás könnyebb megértése érdekében megismételjük az α_t, β_t folyamatoknak azt a modellálását, amely a [7] dolgozatban szerepel. Szorítkozhatunk csak az α_t folyamatra, mert a két folyamat modelljeiben csupán a paraméterek különböznek egymástól. Legyen λ olyan valós szám, melyre $0 \leqq \lambda \leqq 1$ és legyenek $t_1, \ldots, t_n$, továbbá $\tau_1, \ldots, \tau_{n-1}$ független minták, melyeket a (0, 1) intervallumban egyenletes eloszlású sokaságból választottunk. Rendezzük nagyság szerint a $\tau_1, \ldots, \tau_{n-1}$ mintát és jelölje τ_k^* a nagyság szerint k-adikat lentről számítva, tehát $\tau_1^* \leqq \tau_2^* \leqq \ldots \leqq \tau_{n-1}^*$. Vezessük be még a $\tau_0^* = 0$, $\tau_n^* = 1$ jelöléseket, és értelmezzük az α_t valószínűségi változót a következő módon:

$$\alpha_t = cT\lambda \frac{v}{n} + cT(1-\lambda)\tau_v^*, \quad 0 \leqq t \leqq T, \tag{1.2}$$

ahol v azoknak a t_i mintaelemeknek a száma, amelyek kisebbek, mint t; c pozitív állandó, cT egyenlő a vizsgált időszak anyagmennyiségével, mely feltétel szerint egyenlő az összes szállított mennyiséggel. Ha $\lambda = 1$, akkor α_t a $t_1, \ldots, t_n$ minta empirikus eloszlásfüggvénye. A β_t folyamat modelljében n helyett m, λ helyett pedig μ szerepel.

A [7] dolgozatban említés történik az alábbi határértékreláció fennállásáról

$$\lim_{\substack{m\to\infty \\ n\to\infty}} P\left(\sqrt{\frac{mn}{m+n+m(1-\lambda)^2+n(1-\mu)^2}} \sup_{0\leqq t\leqq 1} (\alpha_t - \beta_t) < y\right) = \tag{1.3}$$

$$= \lim_{\substack{m\to\infty \\ n\to\infty}} P\left(\sqrt{\frac{mn}{m+n+m(1-\lambda)^2+n(1-\mu)^2}} \sup_{0\leqq t\leqq 1} (\beta_t - \alpha_t) < y\right) =$$

$$= \begin{cases} 1-e^{-2y^2}, & \text{ha} \quad y > 0, \\ 0, & \text{ha} \quad y \leqq 0, \end{cases}$$

ahol α_t, β_t a fent említett sztochasztikus folyamatok, T értékét egyszerűség kedvéért 1-gyel egyenlőnek választottuk. Ez nyilván nem jelenti az általánosság megszorítását.

Ha az (1.3) relációt közelítőleg érvényesnek vesszük, vagyis n és m elegendően nagyok, akkor adott ε esetén a megbízhatósági egyenlet megoldásaként a következő szám adódik (csak a λ, μ paraméterektől való függést tüntetjük fel az indexben):

$$M_{\lambda,\mu} = c\left[\frac{1}{2}\left(\frac{1+(1-\lambda)^2}{n} + \frac{1+(1-\mu)^2}{m}\right)\log\frac{1}{\varepsilon}\right]^{1/2}. \tag{1.4}$$

Ha α_t determinisztikus és $\alpha_t = ct$ $(0 \leqq t \leqq 1)$, akkor a megfelelő M értéket az (1.4) formulából oly módon kapjuk meg, hogy elvégezzük az $n \to \infty$ határátmenetet. Hasonlóan járunk el a $\beta_t = ct$ $(0 \leqq t \leqq 1)$ esetben. Az előbbi a folytonos és állandó intenzitású anyagbeáramlás, az utóbbi a folytonos és állandó intenzitású anyagfelhasználás esete. Megjegyezzük, hogy λ az egy anyagszállítás alkalmával minimálisan szállítandó δ mennyiséggel a következő összefüggésben van: $\lambda = n\delta/c$. Hasonló a helyzet a β_t folyamat μ paraméterével kapcsolatban.

2. A szállítási és felhasználási folyamat általánosítása

Ebben a szakaszban megismételjük a szállítási folyamatnak a [11] dolgozatban közölt általánosabb modelljét. Minthogy teljesen hasonlóan modellálható a felhasználási folyamat, az utóbbival nem foglalkozunk részletesen.

Az előző szakaszban a szállítási folyamattal kapcsolatban feltételeztük, hogy

a) a szállítási időpontok száma fix, ezt n-nel jelöltük;

b) az n időpont oly módon helyezkedik el az időtengely $(0, 1)$ intervallumában, mint n db egymástól független, egyenletes eloszlású véletlenszerűen választott pont;

c) a teljes szállított mennyiség konstans és egyenlő c-vel, ami egyben a $(0, 1)$ időintervallumbeli anyagfelhasználás; feltehetjük, hogy $c=1$, ez az egység alkalmas megválasztásával mindig elérhető;

d) az egyes szállítási időpontokban a szállított mennyiségek vektora sztochasztikusan független az érkezési időpontok rendszerétől;

e) a szállított mennyiségek modellje a következő: δ jelöli a legkisebb, egy alkalommal biztosan szállított mennyiséget $(0 \leqq \delta \leqq 1/n)$, a fennmaradó $1-n\delta$ mennyiséget pedig oly módon osztjuk el az n időpont között, hogy a $(0, 1-n\delta)$ intervallumot $n-1$ db véletlenszerűen, egymástól függetlenül és egyenletes eloszlás szerint választott pont segítségével felosztjuk n részre, majd a kapott részek által képviselt mennyiségeket hozzárendeljük az egyes időpontokhoz.

Az alábbiakban megtartjuk az *a)*, *c)*, *d)* feltételt, és módosítjuk a *b)*, *e)* feltételeket.

A szállítandó mennyiségek modellálásához a $(0, 1-n\delta)$ intervallumban választunk L db pontot egymástól függetlenül és egyenletes eloszlás szerint, ahol $L > n-1$. Jelöljék $y_1^*, \ldots, y_L^*$ a kapott pontokból alkotott rendezett minta elemeit, $y_1^* \leqq y_2^* \leqq \ldots \leqq y_L^*$. A rendezett mintából kiválasztjuk a $k_1 < k_2 < \ldots < k_{n-1}$ sorszámúakat, végül a δ-val egyenlő fix szállítási mennyiségekhez rendre hozzáadjuk az

$$\eta_1 = y_{k_1}^*, \ \eta_2 = y_{k_2}^* - y_{k_1}^*, \ \ldots, \ \eta_n = 1 - n\delta - y_{k_{n-1}}^* \tag{2.1}$$

mennyiségeket. Ily módon tehát az n alkalommal szállított véletlen mennyiségeket a következő valószínűségi változók képviselik:

$$\delta + \eta_1, \ \delta + \eta_2, \ \ldots, \ \delta + \eta_n. \tag{2.2}$$

Hasonlóan modelláljuk a szállítási időpontok folyamatát. A δ fix mennyiségnek megfelel egy γ minimális idő, mely két szomszédos szállítás között, illetve az első szállítás előtt biztosan van $(0 \leqq \gamma \leqq 1/n)$. Az n db szállítási időpontot egy a $(0, 1-n\gamma)$ intervallumban egyenletes eloszlású sokaság N elemű, $x_1^* \leqq x_2^* \leqq \ldots \leqq x_N^*$ rendezett mintájából származtatjuk oly módon, hogy kiválasztjuk a $j_1 < j_2 < \ldots < j_n$ indexűeket, majd megalkotjuk a

$$\xi_1 = x_{j_1}^*, \ \xi_2 = x_{j_2}^* - x_{j_1}^*, \ \ldots, \ \xi_n = x_{j_n}^* - x_{j_{n-1}}^* \tag{2.3}$$

valószínűségi változókat, és végül a

$$\gamma + \xi_1, \ \gamma + \xi_2, \ \ldots, \ \gamma + \xi_n \tag{2.4}$$

valószínűségi változók részletösszegeiből megalkotjuk az n db szállítási időpontot.

Az $\eta_1, \ldots, \eta_{n-1}$ valószínűségi változók együttes sűrűségfüggvényét $s(z_1, \ldots, z_{n-1})$ jelöli. Ennek képlete igen egyszerű megfontolások alapján a következő (lásd [15]):

$$(2.5)\quad s(z_1, \ldots, z_{n-1}) = \left(\frac{1}{1-n\delta}\right)^n \frac{\Gamma(L+1)}{\Gamma(k_1)\Gamma(k_2-k_1)\ldots\Gamma(k_{n-1}-k_{n-2})\Gamma(L+1-k_{n-1})} \times$$
$$\times \left(\frac{z_1}{1-n\delta}\right)^{k_1-1} \left(\frac{z_2}{1-n\delta}\right)^{k_2-k_1-1} \ldots \left(\frac{z_{n-1}}{1-n\delta}\right)^{k_{n-1}-k_{n-2}-1} \left(1-\frac{z_1+\ldots+z_{n-1}}{1-n\delta}\right)^{L-k_{n-1}},$$

ha $z_i>0$, $i=1, \ldots, n-1$, $z_1+\ldots+z_{n-1}<1-n\delta$ és $s(z_1, \ldots, z_{n-1})=0$ egyébként.

Hasonló képlet adja meg a $\xi_1, \xi_2, \ldots, \xi_n$ valószínűségi változók együttes sűrűségfüggvényét, melyet $r(z_1, \ldots, z_n)$ jelöl:

$$(2.6)\quad r(z_1, \ldots, z_n) = \left(\frac{1}{1-n\gamma}\right)^{n+1} \frac{\Gamma(N+1)}{\Gamma(j_1)\Gamma(j_2-j_1)\ldots\Gamma(j_n-j_{n-1})\Gamma(N+1-j_n)} \times$$
$$\times \left(\frac{z_1}{1-n\gamma}\right)^{j_1-1} \left(\frac{z_2}{1-n\gamma}\right)^{j_2-j_1-1} \ldots \left(\frac{z_n}{1-n\gamma}\right)^{j_n-j_{n-1}-1} \left(1-\frac{z_1+\ldots+z_n}{1-n\gamma}\right)^{N-j_n},$$

ha $z_i>0$, $i=1, \ldots, n$, $z_1+\ldots+z_n<1-n\gamma$ és $r(z_1, \ldots, z_n)=0$ egyébként.

A fentiek szerint az $(\eta_1, \ldots, \eta_{n-1})$ és a $(\xi_1, \ldots, \xi_n)$ valószínűségi vektorváltozók *Dirichlet eloszlásúak*. Ennek a többdimenziós valószínűségeloszlásnak a tulajdonságaival részletesen foglalkozik WILKS [15] könyve.

3. A készletmodellek

I. Modell [11]. Az anyagbeáramlás folyamatát az előző szakaszban leírtaknak megfelelően modelláljuk, és az ottani jelöléseket is megtartjuk. Az anyagfelhasználást időben egyenletes intenzitásúnak tekintjük. Az induló készletet M jelöli. Egyelőre egy anyagfajtával foglalkozunk. Ahhoz, hogy a teljes $(0, 1)$ intervallumban a folyamatos anyagfelhasználást az induló készlet és a beáramló anyag minden időpontban biztosítsa, szükséges és elegendő, hogy teljesüljenek az alábbi relációk

$$(3.1)\quad \begin{aligned} &M && \geqq \gamma+\xi_1,\\ &M+\delta+\eta_1 && \geqq 2\gamma+\xi_1+\xi_2,\\ &M+2\delta+\eta_1+\eta_2 && \geqq 3\gamma+\xi_1+\xi_2+\xi_3,\\ &&& \vdots\\ &M+(n-1)\delta+\eta_1+\eta_2+\ldots+\eta_{n-1} && \geqq n\gamma+\xi_1+\xi_2+\ldots+\xi_n. \end{aligned}$$

Vezessük be a következő jelöléseket

$$(3.2)\quad \begin{aligned} \zeta_1 &= \xi_1,\\ \zeta_2 &= \xi_1+\xi_2-\eta_1,\\ \zeta_3 &= \xi_1+\xi_2+\xi_3-\eta_1-\eta_2,\\ &\vdots\\ \zeta_n &= \xi_1+\xi_2+\ldots+\xi_n-\eta_1-\eta_2-\ldots-\eta_{n-1}. \end{aligned}$$

Az $\eta=(\eta_1, \ldots, \eta_{n-1})$, $\xi=(\xi_1, \ldots, \xi_n)$ valószínűségi vektorváltozók függetlenek, sűrűségfüggvényeik logkonkáv függvények az R^{n-1}, illetve az R^n terekben. Ebből következik, hogy az e két valószínűségi vektorváltozóban foglalt összesen $2n-1$ számú valószínűségi változó együttes sűrűségfüggvénye (a két sűrűségfüggvény szorzata) logkonkáv függvény az R^{2n-1} térben.

Az R^m tér mérhető részhalmazain értelmezett P valószínűségi mértéket logkonkávnak nevezzük (lásd [8]), ha az R^m tér bármely A, B konvex részhalmazai és bármely $0<\lambda<1$ szám esetén fennáll az alábbi egyenlőtlenség

$$P(\lambda A+(1-\lambda)B) \geqq [P(A)]^{\lambda}[P(B)]^{1-\lambda}. \tag{3.3}$$

Az [8] dolgozat alaptétele kimondja, hogy ha egy valószínűségi mértéket logkonkáv valószínűségi sűrűségfüggvény származtat, akkor maga a valószínűségi mérték is logkonkáv. Másfelől [6, Theorem 3] logkonkáv együttes eloszlással bíró valószínűségi változók tetszőleges lineáris transzformáltjai is logkonkáv együttes eloszlással bírnak. Ebből következik, hogy a $\zeta=(\zeta_1, \ldots, \zeta_n)$ valószínűségi vektorváltozó logkonkáv eloszlású.

Az 1. szakaszban említett megbízhatósági egyenlet általánosításaként felírhatjuk az alábbi, M-re vonatkozó egyenletet

$$h(M) = P(\zeta_i \leqq M+(i-1)\delta-i\gamma,\ i = 1, \ldots, n) = p, \tag{3.4}$$

ahol $0<p<1$, de a gyakorlatban $p \approx 1$. Ez most véletlen szállítási folyamat és az állandó intenzitású felhasználási folyamat esetére vonatkozik. Nem okoz azonban nehézséget a véletlen felhasználás esetének hasonló jellegű tárgyalása. A $h(M)$ függvény logkonkáv a teljes számegyenesen, ugyanis logkonkáv eloszlású valószínűségi változó eloszlás-függvénye logkonkáv függvény [8]. Az I. modell több, egymástól függetlenül beáramló anyagfajtára vonatkozik, és a következő nemlineáris programozási feladatban áll

minimalizálandó $(d^{(1)}M^{(1)}+\ldots+d^{(t)}M^{(t)})$,

feltéve, hogy

$$h(\mathbf{M}) = h_1(M^{(1)}) \cdots h_t(M^{(t)}) \geqq p, \tag{3.5}$$

$$\mathbf{M} \geqq \mathbf{0}, \quad \mathbf{M} \in D,$$

ahol $\mathbf{M}=(M^{(1)}, \ldots, M^{(t)})$ és D egy R^t-beli, lineáris (esetleg nemlineáris) feltételekkel meghatározott halmaz. Pl. előírhatjuk, hogy az $\mathbf{M}$ vektor komponensei bizonyos korlát alatt maradjanak, továbbá előírhatjuk, hogy az induló készletek ne foglaljanak el több helyet, vagy ne kössenek le több pénzt egy előírtnál stb. A $d^{(1)}, \ldots, d^{(t)}$ számok nemnegatívak, és az egyes anyagfajták egységnyi készleteinek bizonyos értékelései, melyeket a lokális viszonyok szabnak meg. Elképzelhető a $d^{(1)}M^{(1)}+\ldots+d^{(t)}M^{(t)}$ lineáris célfüggvény helyett nemlineáris célfüggvény használata is.

II. Modell. Ez a modell az előzőtől abban különbözik, hogy a valószínűségi jellegű megszorítás mellett még egy feltételes várható értékekre támaszkodó megszorítással is élünk. Ezáltal nem csupán azt írjuk elő, hogy ritkán forduljon elő anyaghiány, hanem azt is előírjuk, hogy a hiány nagyságának az átlaga ne haladjon meg egy (anyagfajtától és időponttól esetleg függő) szintet. Feltesszük, hogy anyaghiány esetén az anyagigény nem vész el.

A (3.1) egyenlőtlenségek közül ha valamelyik nem teljesül, akkor ez azt jelenti, hogy az adott szállítást közvetlenül megelőzően már anyaghiány volt. Amilyen mértékű az egyenlőtlenség nem teljesülése, olyan mértékű az anyaghiány, pontosabban, olyan hosszú ideje nincs már anyag az adott szállítást megelőzően. Most vettük figyelembe azt a feltételünket, hogy az anyagigény hiány esetén nem vész el. Az egyes anyagfajtákkal kapcsolatos változókra és állandókra felső indexekkel utalunk. Modellünk a következő

minimalizálandó $(d^{(1)}M^{(1)}+\ldots+d^{(t)}M^{(t)})$,

feltéve, hogy

$$h(\mathbf{M}) = h_1(M^{(1)})\cdots h_t(M^{(t)}) \geqq p,$$

$$E\big(\zeta_i^{(j)} - M^{(j)} - (i-1)\delta^{(j)} + i\gamma^{(j)} \mid \zeta_i^{(j)} - M^{(j)} - (i-1)\delta^{(j)} + i\gamma^{(j)} > 0\big) \leqq g_i^{(j)},$$

(3.6)

$$i = 1, \ldots, n; \quad j = 1, \ldots, t,$$

$$\mathbf{M} \geqq \mathbf{0}, \quad \mathbf{M} \in D,$$

ahol a $g_i^{(j)}$ számok általunk előírt állandók, E pedig a várható érték jele. A feltételes várható értékre tett megszorítás helyettesítheti is a valószínűségi jellegű megszorítást. A $\zeta_i^{(j)}$ valószínűségi változó logkonkáv sűrűségfüggvénnyel bír minden i, j esetén. Ebből következik [10], hogy minden i és j esetén a feltételes várható értéket tartalmazó megszorítás bal oldalán az $M^{(j)}$ változó monoton csökkenő függvénye áll, és ilyenformán ez konvertálható az $M^{(j)}$-re tett egyszerű alsó korláttá a [10] dolgozatban általános formában leírt módon. Ennek a jelen esetre való specializálásával a szakasz végén foglalkozunk.

III. Modell. Ismét a megmaradó igény esetével foglalkozunk. Modellünk a (3.6) modelltől abban különbözik, hogy a célfüggvényt kiegészítjük egy büntetés jellegű additív taggal, mely az anyaghiányt hivatott büntetni. Vezessük be a következő valószínűségi változókat

$$\beta_i^{(j)} = \begin{cases} q_i^{(j)}\big(\zeta_i^{(j)} - M^{(j)} - (i-1)\delta^{(j)} + i\gamma^{(j)}\big), & \text{ha } \zeta_i^{(j)} - M^{(j)} - (i-1)\delta^{(j)} + i\gamma^{(j)} > 0 \\ 0 & \text{egyébként,} \end{cases} \tag{3.7}$$

$i=1, \ldots, n$; $j=1, \ldots, t$, ahol $q_i^{(j)} \geqq 0$ minden i és j esetén. A szokásos módon megmutatható, hogy a $\beta_i^{(j)}$ valószínűségi változó $E(\beta_i^{(j)})$ várható értéke az $M^{(j)}$ változó konvex függvénye. Ennek bizonyításához nincs is szükség arra, hogy $\zeta_i^{(j)}$ logkonkáv sűrűségfüggvénnyel bír, elegendő az, hogy van sűrűségfüggvénye. Modellünk a (3.6) modelltől csupán a célfüggvényben különbözik.

Mostani célfüggvényünk a következő

$$\sum_{j=1}^{t} d^{(j)} M^{(j)} + \sum_{j=1}^{t} \sum_{i=1}^{n} E(\beta_i^{(j)}). \tag{3.8}$$

A három modell konstrukciója igazodik ahhoz az általános modelltípushoz, amely a [10] dolgozatban megtalálható.

A III. modell speciális esetként tartalmazza az I. és II. modelleket. A II. modellt a $q_i^{(j)}=0$, $i=1, \ldots, n$; $j=1, \ldots, t$ választással, az I. modellt ugyanezzel és a $g_i^{(j)}=\infty$, $i=1, \ldots, n$; $j=1, \ldots, t$ választással kapjuk meg.

A III. modellnek vannak más, gyakorlatilag szintén számításba jövő fontos speciális esetei. Választhatjuk pl. a p valószínűséget 0-nak és a $q_i^{(j)}$ büntető szorzókat is 0-nak. Ekkor olyan modellt kapunk, mely az I. modellbeli bonyolult valószínűségi megszorítás helyett matematikailag egyszerűbb, feltételes várható értékes megszorításokat tartalmaz. Vagy megtehetjük azt is, hogy a p valószínűséget 0-nak választjuk, a $g_i^{(j)}$ felső korlátokat pedig ∞-nek. Ekkor egy költségjellegű célfüggvényt minimalizálunk az $\mathbf{M} \geqq \mathbf{0}$, $\mathbf{M} \in D$ feltételek mellett, és semmilyen megbízhatósági jellegű megszorításunk nincs a rendszerrel kapcsolatban. E további modelleknek azért nem adunk számot, mert legérdekesebbeknek az I., II., III. modelleket tartjuk.

Szólnunk kell néhány szót a (3.6) feladatban szereplő feltételes várható értékekről és az $E(\beta_i^{(j)})$ várható értékekről. Jelöljék $f_i^{(j)}(x)$ és $F_i^{(j)}(x)$ a $\zeta_i^{(j)}$ valószínűségi változó sűrűség-, illetve eloszlásfüggvényét. A j felső indexet azonban a továbbiakban egyszerűség kedvéért elhagyjuk. Könnyű belátni, hogy általában egy folytonos eloszlású ζ valószínűségi változó és egy a állandó esetén fennáll az

$$E(\zeta - a \mid \zeta - a > 0) = \frac{\int_a^\infty [1 - F(x)]\,dx}{1 - F(a)} = \frac{\int_a^\infty x f(x)\,dx}{1 - F(a)} - a \tag{3.9}$$

egyenlőség, ahol $F(x)$ a ζ valószínűségi változó eloszlásfüggvénye. Ezt alkalmazva azt kapjuk, hogy

$$E(\zeta_i - M - (i-1)\delta + i\gamma \mid \zeta_i - M - (i-1)\delta + i\gamma > 0) = \tag{3.10}$$

$$= \frac{1}{1 - F_i(M + (i-1)\delta - i\gamma)} \int_{M+(i-1)\delta - i\gamma}^{1-n\gamma} [1 - F_i(x)]\,dx =$$

$$= \frac{1}{1 - F_i(M + (i-1)\delta - i\gamma)} \int_{M+(i-1)\delta - i\gamma}^{1-n\gamma} x f_i(x)\,dx - M - (i-1)\delta + i\gamma.$$

Hasonlóan kapjuk, hogy

$$E(\beta_i) = q_i \int_{M+(i-1)\delta - i\gamma}^{1-n\gamma} [1 - F_i(x)]\,dx. \tag{3.11}$$

Egyszerű megfontolással adódik, hogy

$$f_i(x) = \frac{\Gamma(N+1)\Gamma(L+1)}{\Gamma(j_i)\Gamma(N-j_i+1)\Gamma(k_{i-1})\Gamma(L-k_{i-1}+1)} \frac{1}{(1-n\delta)(1-n\gamma)} \times \tag{3.12}$$

$$\times \int_0^{\min(1-n\gamma-x,\, 1-n\delta)} \left(\frac{x+u}{1-n\gamma}\right)^{j_i-1} \left(1 - \frac{x+u}{1-n\gamma}\right)^{N-j_i} \left(\frac{u}{1-n\gamma}\right)^{k_{i-1}-1} \left(1 - \frac{u}{1-n\gamma}\right)^{L-k_{i-1}}$$

ha $0 < x < 1 - n\gamma$ és $f_i(x) = 0$ egyébként, $i = 2, \ldots, n$, továbbá

$$f_1(x) = \frac{\Gamma(N+1)}{\Gamma(j_1)\Gamma(N-j_1+1)} \frac{1}{1-n\gamma} \left(\frac{x}{1-n\gamma}\right)^{j_1-1} \left(1 - \frac{x}{1-n\gamma}\right)^{N-j_1}, \tag{3.13}$$

ha $0 < x < 1 - n\gamma$ és $f_1(x) = 0$ egyébként.

A (3.6) feladat második feltételcsoportjában adott i és j esetén a baloldalon álló kifejezés az $M^{(j)}$ változó folytonos és monoton csökkenő függvénye azoknak az $M^{(j)}$ értékeknek az intervallumán, amelyekre a feltétel valószínűsége pozitív. Emiatt a korlátozó feltétel az $M^{(j)} \geqq M_i^{(j)}$ alakba írható, ahol $M_i^{(j)}$ az az érték, amellyel a feltételi egyenlőtlenség egyenlőséggel teljesül. Az $M_i^{(j)}$ értékeket numerikus integráció segítségével határozzuk meg.

4. A feladatok megoldásai

Ebben a szakaszban megadjuk az ismertetett modellekben szereplő nemlineáris programozási problémák megoldásának egy gyakorlatilag bevált általános módszerét. Csupán az I. modellel foglalkozunk, mert az erre vonatkozó algoritmust kisebb módosítással alkalmazhatjuk a II. és a III. modellre.

Egyszerűség kedvéért az $\mathbf{M} \in D$ feltétel jelentse a következőt: $M^{(j)} \leqq 1$, $j=1, \ldots, t$. Ez különben a feladat szempontjából nem jelent megszorítást, ugyanis triviálisan teljesülnek az alábbi egyenlőségek:

$$h_j(1) = 1, \quad j = 1, \ldots, t, \tag{4.1}$$

amiből következik, hogy az $M^{(j)} \leqq 1$, $j=1, \ldots, t$ feltétel nélkül is az optimális $M^{(1)}, \ldots, M^{(t)}$ értékek mind automatikusan 1-nél kisebbnek adódnak. Az $M^{(j)}$ értékek felülről való korlátozásának csupán annyi jelentősége van, hogy az alkalmazandó SUMT módszer konvergenciájára minden további nélkül hivatkozhatunk.

A SUMT belső pont algoritmusát [2] alkalmazzuk. Tekintsük ehhez az alábbi büntető függvényt

$$G(r, \mathbf{M}) = \sum_{j=1}^{t} d^{(j)} M^{(j)} - r \log \left(\prod_{j=1}^{t} h_j(M^{(j)}) - p \right), \tag{4.2}$$

ahol r rögzített pozitív szám. Könnyen belátható, hogy minden rögzített p esetén $h_1(M^{(1)}) \ldots h_t(M^{(t)}) - p$ is logkonkáv függvény, amiből következik, hogy rögzített r esetén az $\mathbf{M}$ vektorváltozó $G(r, \mathbf{M})$ függvénye konkáv az $\{\mathbf{M} | \mathbf{M} \geqq \mathbf{0}\}$ halmazon. Számunkra ebből csupán az $\{\mathbf{M} | 0 \leqq M^{(j)} \leqq 1, j=1, \ldots, t\}$ halmazon való konkávitás lényeges. A SUMT belső pont módszere mármost oly módon működik, hogy egy 0-hoz konvergáló $r_1 > r_2 > \ldots$ sorozat (elvben) minden r_k eleméhez elvégezzük a $G(r_k, \mathbf{M})$ függvény minimalizálását; e minimum-értékek sorozata konvergál az I. modell minimális célfüggvényértékéhez, tehát elég nagy k esetén a $G(r_k, \mathbf{M})$ függvényt minimalizáló $\mathbf{M}_k$ közelítőleg optimális megoldása feladatunknak.

Korlátos halmaz továbbá folytonos feltételi és célfüggvények esetén a SUMT belső pont algoritmusa konvergál az előbbi értelemben [2], feltéve, hogy a megengedett megoldásokat meghatározó egyenlőtlenségek mind határozott egyenlőtlenséggel teljesülnek a belső pontokban. Ami az I. modellt illeti, a $0 \leqq M^{(j)} \leqq 1$, $j=1, \ldots, t$ feltételek már az ugyanezen feltételekkel meghatározott t-dimenziós egységkocka belső pontjaiban is határozott egyenlőtlenséggel teljesülnek, elegendő tehát a valószínűségi feltétellel foglalkozni. Legyen $\mathbf{M}_1$ a megengedett megoldások halmazának belső pontja. Megmutatjuk, hogy $h(\mathbf{M}_1) > p$, ahol $0 < p < 1$. Az $\mathbf{1}$ és az $\mathbf{M}_1$ vektorokat összekötő szakasz benne van a megengedett megoldások halmazában, minthogy ez konvex halmaz. E szakaszt $\mathbf{M}_1$-ből tovább meghosszab-

bítva, a kapott félegyenesen válasszunk egy $\mathbf{M}_0 \neq \mathbf{M}_1$ megengedett megoldást. Ekkor alkalmas $0<\lambda<1$ számmal fennáll az

$$\mathbf{M}_1 = \lambda \mathbf{1} + (1-\lambda)\mathbf{M}_0$$

egyenlőség, amiből $h(\mathbf{M})$ logkonkávitása miatt adódik, hogy

$$h(\mathbf{M}_1) \geqq [h(\mathbf{1})]^{\lambda}[h(\mathbf{M}_0)]^{1-\lambda} \geqq p^{1-\lambda} > p. \tag{4.3}$$

Ezzel tehát bebizonyítottuk, hogy a SUMT belső pont algoritmus konvergens a mi esetünkben.

Ami a (4.2) függvény feltétel nélküli minimalizálását illeti, erre sok általános módszer specializálható. A feltétel nélküli minimalizálási eljárások egy része gradiens mentes, másik része használja a gradienst. Az utóbbi módszerek alkalmazásának megkönnyítése céljából megadjuk a $h(\mathbf{M})$ függvény gradiensének egy viszonylag egyszerű kiszámítási módját. Ez annál is inkább figyelmet érdemel, mert a függvényérték kiszámítási módjához hasonló módszerre most is szükség van. Minthogy a $h(\mathbf{M})$ függvény a $h_j(M^{(j)})$, $j=1, \ldots, t$ függvények szorzata, a gradiens megalkotásakor elegendő külön a $h_j(M^{(j)})$ függvények deriváltjaival foglalkozni. Rövidség kedvéért hagyjuk el az indexeket, és foglalkozzunk a (3.4) egyenlőségben szereplő függvény M szerinti deriváltjával. E függvény tulajdonképpen a $\zeta_1, \ldots, \zeta_i$ valószínűségi változók együttes eloszlásfüggvénye az $M+(i-1)\delta-i\gamma$, $i=1, \ldots, n$ helyeken.

Megjegyezzük, hogy egy n-dimenziós folytonos eloszlás $F(z)$ eloszlásfüggvényére fennáll az alábbi egyenlőség

$$\frac{\partial F(z)}{\partial z_i} = F(z_j, j \neq i | z_i) f_i(z_i), \quad i = 1, \ldots, n, \tag{4.4}$$

ahol $f_1, \ldots, f_n$ az egydimenziós peremeloszlások sűrűségfüggvényei, $F(\cdot \mid \cdot)$ pedig az i-edik változóra tett feltétel melletti $n-1$-dimenziós feltételes eloszlásfüggvény.

Feltesszük, hogy $n\delta<1$, $n\gamma<1$. Ha $n\delta=1$, $n\gamma=1$ közül legalább az egyik teljesül, eljárásunk lényegesen egyszerűsödik. A (3.4) alatti $h(M)$ függvény esetében előbb vesszük a

$$P\big(\zeta_i \leqq z_i+(i-1)\delta-i\gamma,\ i=1, \ldots, n\big) \tag{4.5}$$

n-változós függvény $z_1, \ldots, z_n$ szerinti deriváltjait a $z_1=\ldots=z_n=M$ helyen. Ezek összege adja a $h(M)$ függvény M szerinti deriváltját. A (4.5) függvény z_i szerinti deriváltjára a (4.4) formulát alkalmazzuk, és a deriváltat mindjárt az említett $z_1=\ldots=z_n=M$ helyen vesszük. Eredményként a következőt kapjuk:

$$P\big(\zeta_j \leqq M+(j-1)\delta-j\gamma,\ j \neq i | \zeta_i = M+(i-1)\delta-i\gamma\big) f_i\big(M+(i-1)\delta-i\gamma\big) = \tag{4.6}$$

$$= f_i\big(M+(i-1)\delta-i\gamma\big) \int_0^{\min\{1-M-(i-1)\delta-(n-i)\gamma,\ 1-n\delta\}} P\big(\zeta_j \leqq M+(j-1)\delta-j\gamma,\ j \neq i | \xi_1 +$$

$$+ \ldots + \xi_i = M+(i-1)\delta-i\gamma+x,\ \eta_1+\ldots+\eta_{i-1}=x\big) \times$$

$$\times \frac{\Gamma(N+1)}{\Gamma(j_i)\Gamma(N+1-j_i)} \frac{1}{1-n\gamma} \left(\frac{M+(i-1)\delta-i\gamma+x}{1-n\gamma}\right)^{j_i-1} \left(1-\frac{M+(i-1)\delta-i\gamma+x}{1-n\gamma}\right)^{N-j_i} \times$$

$$\times \frac{\Gamma(L+1)}{\Gamma(k_{i-1})\Gamma(L+1-k_{i-1})} \frac{1}{1-n\delta} \left(\frac{x}{1-n\delta}\right)^{k_{i-1}-1} \left(1-\frac{x}{1-n\delta}\right)^{L-k_{i-1}} dx,$$

ahol $f_i(z)$ a ζ_i valószínűségi változó sűrűségfüggvényét jelöli. A (4.6) kifejezések második sorában álló feltételes valószínűség kifejezhető abszolút valószínűség formájában is, miáltal erre egy a (3.4) alatti valószínűséghez hasonló kifejezésű valószínűséget kapunk más paraméterekkel.

Emlékeztetünk arra, hogy a $\xi_1, \ldots, \xi_n$ valószínűségi változók egy a (0, 1) intervallumból vett N-elemű mintából származnak a 2. szakaszban leírt módon. A $\xi_1+\ldots+\xi_i=u$ $(u=M+(i-1)\delta-i\gamma+x)$ feltétel mellett $\xi_1, \ldots, \xi_n$ együttes eloszlása megegyezik két független valószínűségi vektorváltozó együttes eloszlásával. E véletlen vektorok j_i-1, illetve $N-j_i$ komponensből állnak, és az együttes sűrűségfüggvény mindkét esetben egy (2.6) típusú képlettel adható meg, miközben N, n, $1-n\gamma$ helyét az első vektor esetében j_i-1, $i-1$, u, a második vektor esetében pedig $N-j_i$, $n-i$, $1-n\gamma-u$ veszik át. Hasonló a helyzet az $\eta_1, \ldots, \eta_{n-1}$ valószínűségi változókkal.

A $h(\mathbf{M})$ valószínűség kiszámítására szimulációs eljárást alkalmazunk. A gradiens kiszámítására előbb javasolt módszer ennél bonyolultabb, mert még numerikus integrációt is igényel a szimuláció több esetben való végrehajtásán túl. Célszerűnek látszik tehát a feltétel nélküli minimalizáláskor a gradiensmentes módszerekhez folyamodni.

5. A $h(\mathbf{M})$ függvény értékeinek meghatározására vonatkozó szimulációs eljárás

A $h(\mathbf{M})$ valószínűség egyben feltételi függvényérték is. Ennek a kiszámítása adott $\mathbf{M}$ esetében módszerünkben szimulációval történik. A szimuláció végrehajtására két módszer is kínálkozik. Az egyik pontosan követi a szállítási időpontok és a szállított mennyiségek modellálását: N, illetve L elemű mintákat választunk nagy számban, ezeket rendezzük, és a kívánt sorszámú elemeket kiválasztjuk stb. Ennek a módszernek nagy hátránya, hogy a rendezés műveletének végrehajtása igen sok gépidőt igényel. Bár jól szervezett rendező rutinnal sikerül viszonylag jó gépidőt elérni, mégis nagy N (illetve L), esetén mindenképpen kedvezőtlen a módszer. Ismeretes ugyanis, hogy N elem rendezésének időtartama legalább $N \log N$ nagyságrendben növekszik [4].

A másik, az előbbinél hatékonyabb szimulációs módszer azon alapszik, hogy tetszőleges *Dirichlet eloszlás* reprezentálható (lásd [15]), mint olyan $y_1, \ldots, y_n$ valószínűségi változók eloszlása, melyek

$$y_i = \frac{x_i}{x_1+\ldots+x_{n+1}}, \quad i = 1, \ldots, n \tag{5.1}$$

alakúak, ahol $x_1, \ldots, x_{n+1}$ független, standard gamma eloszlású valószínűségi változók, azaz x_i sűrűségfüggvénye az alábbi

$$\frac{z^{\vartheta_i-1}e^{-z}}{\Gamma(\vartheta_i)}, \quad z>0; \tag{5.2}$$

$\vartheta_1, \ldots, \vartheta_{n+1}$ pozitív állandók. Valóban, az (5.1) valószínűségi változók együttes sűrűségfüggvénye a következő

$$(5.3) \qquad \frac{\Gamma(\vartheta_1+\ldots+\vartheta_{n+1})}{\Gamma(\vartheta_1)\ldots\Gamma(\vartheta_{n+1})} z_1^{\vartheta_1-1}\ldots z_n^{\vartheta_n-1}(1-z_1-\ldots-z_n)^{\vartheta_{n+1}-1},$$

ha $z_i>0$, $i=1, \ldots, n$, $z_1+\ldots+z_n<1$, egyébként pedig zéró, tehát $\vartheta_1, \ldots, \vartheta_{n+1}$ alkalmas megválasztásával a kívánt *Dirichlet eloszlás* áll elő.

AHRENS és DIETER [1] hatékony módszert dolgozott ki gamma eloszlású valószínűségi változók szimulálására. Ez a módszer akkor igen előnyös, amikor a paraméter nagy, vagy nem egész szám.

A (2.5) és a (2.6) sűrűségfüggvények kissé eltérnek az (5.3) sűrűségfüggvénytől. A szimulációra vonatkozó előbb említett eljárás csak egyszerű módosítást igényel mindkét esetben. Tekintsük a gamma eloszlás sűrűségfüggvényét:

$$(5.4) \qquad \frac{\lambda^{\vartheta} z^{\vartheta-1} e^{-\lambda z}}{\Gamma(\vartheta)}, \quad z>0.$$

Ha $x_1, \ldots, x_{n+1}$ független, rendre $\lambda, \vartheta_1; \ldots; \lambda, \vartheta_{n+1}$ paraméterekkel bíró gamma eloszlású valószínűségi változók, ahol $\lambda=1-n\gamma$, $\vartheta_1=j_1$, $\vartheta_2=j_2-j_1$, $\ldots$, $\vartheta_n=j_n-j_{n-1}$, $\vartheta_{n+1}=N-j_n+1$, akkor az (5.1) képlet szerint származtatott $y_1, \ldots, y_n$ valószínűségi változók együttes eloszlása megegyezik $\xi_1, \ldots, \xi_n$ együttes eloszlásával. Az x_i valószínűségi változó viszont előállítható mint ϑ_i számú független, azonos, λ paraméterű exponenciális eloszlású valószínűségi változó összege, $i=1, \ldots, n+1$. Végül az exponenciális eloszlású valószínűségi változók a (0, 1) intervallumban egyenletes eloszlású valószínűségi változók negatív logaritmusaiként állíthatók elő. Hasonló módon származtatható az $\eta_1, \ldots, \eta_{n-1}$ valószínűségi változók együttes eloszlása.

E második szimulációs eljárás esetében az $N+1$ számú egyenletes eloszlásból vett mintaelemeket csupán logaritmálnunk kell, de nem kell rendeznünk, ezért az első eljárásnál sokkal kedvezőbb gépidő érhető el.

A valószínűségeket a szimuláció révén nyert relatív gyakoriságokkal közelítjük. Az adott pontosságot biztosító mintaelemszámot a *Csebisev egyenlőtlenség Bernstejn-féle élesítésére* [13] támaszkodva határozzuk meg. Az említett *Bernstejn-egyenlőtlenség* a következő: ha v_m jelenti m kísérlet során egy p valószínűségű esemény gyakoriságát, akkor

$$(5.5) \qquad P\left(\left|\frac{v_m}{m}-p\right| \geqq \varepsilon\right) \leqq 2\exp\left[-\frac{m\varepsilon^2}{2p(1-p)\left(1+\frac{\varepsilon}{2p(1-p)}\right)^2}\right].$$

Ha a baloldalon álló valószínűséget δ-val tesszük egyenlővé, akkor m-re azt kapjuk, hogy

$$(5.6) \qquad m \geqq \frac{2p(1-p)\left(1+\frac{\varepsilon}{2p(1-p)}\right)^2}{\varepsilon^2} \log\frac{2}{\delta}.$$

Ha p változik 0 és 1 között, minden más pedig állandó (5.5) jobb oldalán, akkor a legnagyobb érték a $p=1/2$ esetben adódik. Az így kapott alsó határ minden p valószínűségre univerzálisan jó. Ez azért fontos, mert éppen a p értéket nem ismerjük. Ám gyakran vannak p értékére használható korlátaink. A sztohasztikus programozási feladatok többnyire ilyenek, legalábbis akkor, amikor a feladatban valószínűségre vonatkozó alsó korlát is van a feltételek között.

Készletmodelljeinkben a $h(\mathbf{M})$ függvény alsó korlátjaként legalább 0,8 használatos. Ez azt jelenti, hogy a tényezők általában meghaladják a 0,9 szintet. Ilyenformán a szükséges mintaelemszám sokkal kisebb, mintha a valószínűségekre semmilyen információ nem állna rendelkezésre. Az alábbi táblázat jól illusztrálja az m mintaelemszám változását ε és p függvényében, 90% biztonság mellett (a táblázatolt mintaelemszám az a legkisebb m, amelyre (5.5) bal oldala nem nagyobb, mint 0,1):

p \ ε	0,1	0,05	0,025	0,01
0,5	216	726	2646	15606
0,8	165	513	1785	10209
0,9	130	352	1120	6016
0,95	120	265	727	3481

6. Numerikus példa

Példaként egy gyártmány előállításához szükséges két anyagféleség készletezési problémájának megoldását mutatjuk be. Mindkét anyag raktáron tartása költségkihatásaiban jelentős, bár nem egyenlő mértékben (a második egységára az első háromszorosa), ugyanakkor bármelyik hiánya a gyártás leállásához vezet.

A cikket nagy sorozatban gyártják. Mindkét anyagfajta felhasználása a negyedév folyamán jó közelítéssel egyenletesnek tekinthető. A negyedéves anyagszükséglet a termelési terv alapján ismert.

Ha valamelyik anyagból hiány lép fel, a cikk gyártásával le kell állni mindaddig, amíg újabb anyag nem érkezik. Ekkor a tervtől való lemaradást pótolni kell. Tehát a megmaradó igények esetéről van szó, ennek a feltételes várható érték típusú korlátozásnál van nagy jelentősége.

A két anyagfajtát különböző helyről szállítják, a két szállítási folyamat egymástól függetlennek tekinthető. A szállító fél vállalja, hogy a negyedévre megrendelt anyagot a periódus végéig teljes egészében leszállítja, de a szállítás több tételben, véletlenszerű mennyiségekben és időpontokban történik. Az előző periódusok tapasztalata alapján a szállítások száma ismert, az első anyagfajtára 4, a másodikra 5 egy negyedévben. Mindegyik negyedévet 90 naposnak tekintjük.

A szállítási folyamat leírására a korábbi negyedévek tapasztalati adatait használjuk fel. Ezt az első anyagra az alábbiakban részletezzük:

A szállítások napjai a negyedév 1—90 napjain belül

Negyedév \ A szállítás sorszáma	1.	2.	3.	4.
1.	23	41	61	82
2.	27	48	73	88
3.	30	39	60	90
4.	19	48	68	89
5.	24	50	65	78
6.	28	42	71	82
Az oszlopok átlaga:	25,17	44,66	66,33	84,83

Két szomszédos szállítás között a minimális időtartam 9 nap. A negyedév 90 napját a (0, 1) intervallumnak tekintve azt kapjuk, hogy $\gamma^{(1)}=9/90=0{,}1$, a szállítási időpontok empirikus várható értékei pedig:

$$\bar{z}_1 = 0{,}28, \quad \bar{z}_2 = 0{,}49, \quad \bar{z}_3 = 0{,}73, \quad \bar{z}_4 = 0{,}94.$$

A szállítási időpontok korábban leírt modellezése szerint

$$\bar{z}_i = i\gamma^{(1)} + E(x^*_{j_i}) \quad i = 1, 2, \ldots, n, \text{ ahol}$$

$x^*_{j_i}$ jelöli a $(0{,}1-n\gamma^{(1)})$ intervallumból egyenletes eloszlás szerint vett N elemű minta nagyság szerint j_i-edik elemét. Feladatunk, hogy meghatározzuk az N és j_i, $i=1, \ldots, n$ olyan egész értékeit, melyekre teljesülnek a fenti egyenlőségek, azaz:

$$E(x^*_{j_i}) = j_i \frac{1-n\gamma^{(1)}}{N} = z_i - i\gamma^{(1)}, \quad \text{ebből}$$

$$j_i = \frac{\bar{z}_i - i\gamma^{(1)}}{1-n\gamma^{(1)}} N, \quad i = 1, \ldots, n.$$

Racionális $\bar{z}_i$ és $\gamma^{(1)}$ esetén (a gyakorlatban mindig ez a helyzet), van ilyen j_i és N. Az empirikus várható érték pontatlansága és a modellálás jellege miatt azonban nem mindig érdemes a fenti egyenlőséget pontosan kielégítő j_i és N választására törekedni. N nagy értéke esetén a modell számítástechnikai kiértékelése is jelentősen hosszabb időt vesz igénybe. Tapasztalataink alapján N értékének $2n$ és $10n$ közötti választása megfelelő pontosságot ad.

A vizsgált példában $\frac{\bar{z}_i - i\gamma^{(1)}}{1-n\gamma^{(1)}}$, $i=1, 2, 3, 4$, értékei rendre 0,298; 0,493; 0,728; 0,903, így N értékét 10-nek választva, kielégítő pontosságot eredményez a $j_1=3$, $j_2=5$, $j_3=7$, $j_4=9$ választás.

A szállítási tételek nagyságára vonatkozó ismereteinket az alábbi táblázatban foglaljuk össze:

Negyedév \ A szállítás sorszáma	1.	2.	3.	4.	Összesen
1.	630	400	670	800	2500
2.	700	500	600	900	2700
3.	730	580	550	740	2600
4.	720	620	650	1010	3000
5.	760	580	760	1100	3200
6.	750	650	780	920	3100

A negyedéves összmennyiség szerint soronként osztva az újabb táblázat a következő:

Negyedév \ A szállítás sorszáma	1.	2.	3.	4.
1.	0,252	0,16	0,268	0,32
2.	0,259	0,185	0,222	0,333
3.	0,28	0,223	0,211	0,284
4.	0,24	0,206	0,216	0,336
5.	0,237	0,181	0,238	0,344
6.	0,242	0,21	0,252	0,297
Az oszlopok szerinti átlagok:	0,252	0,194	0,234	0,319

Jelöljék ezeket az empirikus várható értékeket rendre $\bar{u}_1, \bar{u}_2$ és $\bar{u}_3$ (az utolsó oszlop csak ellenőrzésül szolgál), és $v_i = \sum_{j=1}^{i} \bar{u}_i$, $i=1, 2, 3$. A minimálisan leszállított $(0, 1)$ intervallumra normált tételnagyság $\delta^{(1)}=0{,}16$.

A szállítási időpontok modellezéséhez hasonlóan eljárva felírjuk az alábbi egyenlőséget:

$$v_i = i\delta^{(1)} + E(y^*_{k_i}), \quad i = 1, \ldots, n-1,$$

ahol $y^*_{k_i}$ jelöli a $(0{,}1-n\delta^{(1)})$ intervallumból egyenletes eloszlás szerint vett L elemű minta nagyság szerint k_i-edik elemét:

$$E(y^*_{k_i}) = k_i \frac{1-n\delta^{(1)}}{L} = v_i - i\delta^{(1)},$$

$$k_i = \frac{v_i - i\delta^{(1)}}{1-n\delta^{(1)}} L, \quad i = 1, \ldots, n-1,$$

ahol az L egész számot úgy kell megválasztani, hogy a k_i, $i=1, \ldots, n-1$ értékek egészek legyenek.

A $\frac{v_i - i\delta^{(1)}}{1-n\delta^{(1)}}$, $i=1, 2, 3$ kifejezés értékei rendre 0,255, 0,35, 0,555. Eszerint az $L=20$ és $k_1=5$, $k_2=7$, $k_3=11$ választás megfelelő közelítést ad.

A másik anyagfajta esetén ugyanígy határoztuk meg az $n=5$, $N=10$, $j_1=2$, $j_2=3$, $j_3=5$, $j_4=7$, $j_5=9$, $\gamma^{(2)}=0{,}15$ $L=10$, $k_1=2$, $k_2=5$, $k_3=7$, $k_4=8$, $\delta^{(2)}=0{,}12$ paraméterértékeket.

A SUMT belső pont algoritmust az $M^{(i)}$ értékek 0,6 és 0,8 közötti választásával indítottuk. A gyakorlati feladatokban ezek általában a megengedett tartomány belsejében vannak. r_1 értékét 1-nek választottuk, és ezt lépésenként 1/5 részére csökkentettük, így már három lépés után a büntető függvény miatt a célfüggvényben adódó eltérés 0,01 alatt van. A (4.2) függvények $r=r_k$ érték melletti feltétel nélküli minimalizálásakor is 0,01 hibakorlátot írtunk elő. Ha a célfüggvény értéke ennél kevesebbet változik a feltétel nélküli minimalizálás valamelyik lépésénél (ez általában $3=4$ lépés után teljesült), az $r=r_{k+1}=r_k/5$ értékkel minimalizáljuk a (4.2) függvényt az előző lépésben adódó végpontból indulva. Az $r=1/125$ értékre adódó minimumot fogadtuk el a feladat megoldásának.

A korábban vázolt numerikus példában $p=0{,}8$ választása mellett $M^{(1)}=0{,}32$ és $M^{(2)}=0{,}29$ adódott megoldásként. Ezt a következő negyedév összes tervezett fogyasztásával beszorozva kapjuk a két anyagból szükséges biztonsági készlet nagyságát.

A megoldott feladatban az eloszlásfüggvények értékét először 10% hibahatárral közelítettük. $M^{(1)}$ és $M^{(2)}$ induló értékeit 1-hez közelinek választottuk, emiatt $h(\mathbf{M})$ is közelítőleg 1. $h(\mathbf{M})$ értékének meghatározásához a ζ valószínűségi vektorváltozót 150-szer generáltuk egymástól függetlenül. Ha a minimalizáló eljárás során a h függvény értéke az előírt p korlátot 0,1-nél jobban megközelítette, akkor értékének pontosítása céljából a mintaelemek számát 400-ra növeltük. Végül 0,05-ös eltérés esetén 1800 elemű mintát vettünk. Egy eloszlásfüggvény-érték meghatározásához ennél több mintaelemet nem generáltunk. A feladatunkban szereplő $h(\mathbf{M})\geqq 0{,}8$ egyenlőtlenség maga után vonja, hogy $h_1(M^{(1)})\geqq 0{,}8$, $h_2(M^{(2)})\geqq 0{,}8$. Ebből következik, hogy a megoldáspontban a tényezők valószínűségértékei legalább 90%-os biztonsággal legfeljebb 2,5%-kal térnek el a szimulációval nyert relatív gyakoriságtól.

A futtatott mintapéldákban az összes szükséges gépidő 1,5 és 2,5 perc között volt. A számításokat FORTRAN nyelven írt programmal CDC 3300-as gépen végeztük. A feltétel nélküli minimalizálásra vonatkozólag több módszerrel kapcsolatban is gyűjtöttünk tapasztalatot. Ezek közül HOOKE és JEEVES [3], ROSENBROCK [14], valamint POWELL [6] módszerét említjük. A gradiens értékét felhasználó eljárások alkalmazása a korábban már vázolt nehézségek miatt nem látszik célszerűnek. Legjobbnak HOOKE és JEEVES módszere bizonyult ami az optimum megtalálásának gyorsaságát illeti. Ez a módszer sikeresnek bizonyult már más esetben is, amikor a függvényértékek kiszámítása szintén szimulációval történt. A tapasztalatok szerint a minimalizáláshoz szükséges gépidő a kezdőpont megválasztásától nem függ lényegesen.

A megoldott mintapéldák gépidő felhasználása és a numerikus eredmények reálissá teszik a modell gyakorlatban való alkalmazhatóságát költségkihatásaiban jelentős cikkek készletpolitikájának, illetve készletnormáinak kialakítására.

IRODALOM

[1] AHRENS, J. H. and DIETER, K., "Computer methods for sampling from gamma, beta, Poisson and binomial distributions,", *Computing* **12** (1974) 223—246.
[2] FIACCO, A. V. and MCCORMICK, G. P., *Nonlinear programming: Sequential unconstrained minimization technique* (Wiley, New York, 1968).
[3] HOOKE, R. and JEEVES, T. A., "Direct search solution of numerical and statistical problems", *Journal of the A.C.M.* **8** (1959) 215—229.
[4] KNUTH D. E., *The art of Computer Programming, Vol. 3, Sorting and Searching* (Addison—Wesley P. C., 1973).
[5] LÁSZLÓ, Z., „Egy teljesen véletlen megbízhatósági jellegű készletmodell", kandidátusi értekezés, Budapest, 1970.
[6] POWELL, M. J. D., "An iterative method for finding the minimum of a function of several variables without calculating derivatives", *Computer Journal* **7** (1964) 155—162.
[7] PRÉKOPA, A., "Reliability equattion for an inventory problem and its asymptotic solutions", in: *Coll. on Appl. of Math. to Economics* (Akadémia Kiadó, Budapest, 1965). 317—327.
[8] PRÉKOPA, A., "On logarithmic concave measures with application to stochastic programming", *Acta Universitatis Szegedienis* **32** (1971) 301—316.
[9] PRÉKOPA, A., "A class of stochastic programming decision problems", *Mathematische Operationsforschung und Statistic* **3** (1972) 349—354.
[10] PRÉKOPA, A., "Contributions tho the theory of stochastic programming", *Mathematical Programming* **4** (1973) 202—221.
[11] PRÉKOPA, A., "Stochastic Programming Models for Inventory Control and Water Storage Problems", in: *Colloquia Mathematica Societatis János Bolyai 7. Invertory Control and Water Storage Győr*, 1971. (Bólyai János Math. Soc. and North Holland Publ. Comp., Budapest, 1973).
[12] PRÉKOPA, A., "Generatizations of the Theorems of Smirnov with Application to a Reliability Type Inventory Problem", *Mathematische Operations-forschung und Statistic* **4** (1973) 283—297.
[13] RÉNYI, A., *Valószínűségszámítás* (Tankönyvkiadó, Budapest, 1966.).
[14] ROSENBROCK, H. H., "An Automatic Method for finding the Greatest or Least Volue of a Function", *The Computer Journal* **3** (1960) 173—184.
[15] WILKS, S. S., *Mathematical Statistics* (Wiley, New York, 1962).
[16] ZIERMANN, M., »Anvendung des Smirnov'schen Sätzen auf Lagerhaltungsproblems«, *Publications of the Mathematical Institut of the HAS* **8** (1965) Series B 509—518.

(Beérkezett: 1976. február 26.)

PRÉKOPA ANDRÁS ÉS KELLE PÉTER
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1502 BUDAPEST XI., KENDE ÚT 13—17.

RELIABILITY TYPE INVENTORY MODELS BASED ON STOCHASTIC PROGRAMMING

A. PRÉKOPA and P. KELLE

The models discussed in this paper are the generalisations of the inventory models of A. PRÉKOPA [7, 12] and M. ZIERMANN [16] which decide the necessary initial stock ensuring the continuous supply for an item with given probability. Now the initial stock of several types of items is to be decided, furthermore the random arrival process of the particular types are not necessarily homogeneous stochastic processes in time. The models to be described are stochastic programming problems such that in this case the initial stock is determined algorithmically and not by formulas like in [7, 12] and [16]. This means the solution of a nonlinear programming problem for every model, in the course of which the function values in the probabilistic constraint are evaluated by simulation technique. A numerical example is given for the illustration of the method.

Alkalmazott Matematikai Lapok **2** *(1976) 17—26*

A TÖBBDIMENZIÓS TÉR HALMAZAI VALÓSZÍNŰSÉGEINEK KISZÁMÍTÁSA NORMÁLIS ELOSZLÁS ESETÉN

DEÁK ISTVÁN
Budapest

A dolgozatban közlünk egy számítógépes algoritmust, amely alkalmas a többdimenziós normális eloszlással kapcsolatos valószínűségek számszerű meghatározására. A közölt technika alapján működő szubrutinok tetszőleges korrelációs mátrix esetén 50 dimenzióig gyakorlatilag elég rövid idő alatt és megfelelő pontossággal számítják ki a többdimenziós normális eloszlás eloszlásfüggvényének a konkrét értékeit. Megadjuk a szubrutinok futására vonatkozó ellenőrző számításokat és az időeredményeket.

1. Bevezetés

A többdimenziós tér halmazai valószínűségeinek kiszámítása a sztochasztikus programozásban és a többváltozós matematikai statisztika módszereiben sok helyen szükséges. Például a PRÉKOPA ANDRÁS által felállított STABIL sztochasztikus programozási modell [21], [22] számítógépes kiszámítása [6] folyamán, amely modell a következő alakú

$$P\{g_i(\mathbf{x}) \geqq \beta_i, \quad i = 1, \ldots, n\} \geqq p,$$

$$g_i(\mathbf{x}) \geqq b_i, \quad i = n+1, \ldots, n+m, \tag{1.1}$$

$$\min f(\mathbf{x}),$$

ahol a $\beta_1, \ldots, \beta_n$ valószínűségi változók együttes eloszlása n dimenziós normális eloszlás is lehet. A többdimenziós tér halmazai valószínűségeinek kiszámítására van szükség PRÉKOPA egyéb sztochasztikus programozási modelljeinek [15]—[20] az optimalizálása folyamán. Többváltozós statisztikai módszerek közül BENE [4] dolgozatára utalunk példaként.

A többdimenziós tér halmazai valószínűségeinek kiszámítása sok esetben az eloszlásfüggvény kiszámítására vezethető vissza. Az általunk közölt algoritmus mind az általános halmazok valószínűségeinek, mind az eloszlásfüggvény konkrét értékeinek kiszámítását képes elvégezni többdimenziós normális eloszlás esetén. A cikkben a többdimenziós normális eloszlás eloszlásfüggvényének a kiszámításával foglalkozunk.

A többdimenziós normális eloszlás eloszlásfüggvényét általában nem lehet pontosan kiszámítani. A két- és a háromdimenziós esetekben vannak táblázatok, magasabb dimenzióban csak speciális korrelációs mátrix esetén vagy egyes kitüntetett pontokban adható meg pontos érték a [10] dolgozat szerint.

A sűrűségfüggvény integrálási határainak transzformálásával egyváltozós esetre vezeti vissza MILTON a normális eloszlásfüggvény kiszámítását $n=6$ dimenzióig

[14]. DUTT az integrandus transzformációját hajtja végre $n=4$ dimenzióig [9]. BENE [4] dolgozatában a sűrűségfüggvény sorbafejtésével (ANDEL [2] cikkében található módhoz hasonlóan) és a sor konvergenciáját gyorsító *Euler-féle sortranszformációkkal* ért el jó eredményeket $n=10$ dimenzióig. JOHNSON és KOTZ [11] könyvében *Monte Carlo integrálásra* található utalás. Szerző saját vizsgálatai a többdimenziós normális eloszlás eloszlásfüggvénye konkrét értékének *Monte Carlo integrálással* való kiszámítására vonatkozólag inkább negatívnak mondható eredményt adtak, amennyiben 8—12 dimenzió felett gyakorlatilag lehetetlenné válik a kiszámítás [7].

Egy *Monte Carlo számítógépes technikát* írunk le, amely többdimenziós normális eloszlású pszeudo-véletlen vektorokat állít elő, és egy ezt alkalmazó szubrutinrendszert, amely a $\Phi(\mathbf{h})$ konkrét értékét számítja ki, ahol $\Phi(\mathbf{h})$ a zérus várható érték vektorú és **1** szórású, **R** korreláció mátrixú n dimenziós normális eloszlásfüggvény, vagyis

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

Az alkalmazott technikát úgy választottuk meg, hogy a számítógép egy konkrét $\Phi(\mathbf{h})$ értéket a lehető legrövidebb időn belül számítson ki, a szubrutinok tetszőleges **h** vektor és R korreláció mátrix esetén működjenek, és minél magasabb n dimenzióig használhatók legyenek. A szubrutinrendszer $n=50$ dimenzióig gyakorlatilag megfelelő rövid idő alatt és a STABIL modell optimalizálási algoritmusa folyamán megengedhető hibával számítja ki a konkrét $\Phi(\mathbf{h})$ értékeket. Egyes értékeket még $n=200$ dimenziós normális eloszlásfüggvény esetén is ki lehet számítani a leírt technika segítségével, amelyek elvileg torzítatlan becslést adnak a valószínűségre.

A 2. szakaszban az alkalmazott *Monte Carlo technikát* írjuk le röviden, a 3. szakaszban a szubrutinok számítógépes ellenőrzését, végül a 4. szakaszban pedig összefoglaljuk a számítógépes tapasztalatokat, és összehasonlítjuk eljárásunkat az irodalomban található eredményekkel.

2. A többdimenziós normális eloszlás eloszlásfüggvényének kiszámítása pszeudo-véletlen vektorok beesési gyakoriságával

Az alkalmazott eljárást három részre bontjuk fel: egydimenziós standard normális eloszlású változókat generálunk, ezekből n dimenziós vektorokat alkotunk, amelyek eloszlásfüggvénye a vizsgált Φ többdimenziós normális eloszlásfüggvény lesz, és végül ezeknek a **h** vektor által meghatározott térrészbe való beesési gyakoriságával közelítjük a keresett $\Phi(\mathbf{h})$ értéket. A dolgozatban csak a **0** várható értékvektorral rendelkező és **1** szórásvektorú eloszlással foglalkozunk, mivel egy lineáris transzformációval könnyen módosíthatjuk a 2. b) alatt leírt vektorokat, ha erre van szükség.

a) Standard normális eloszlású pszeudo-véletlen számok generálása

Sokféle módszer ismeretes normális eloszlású pszeudo-véletlen számok generálására [1]. A számítógépes programban az irodalomban fellelhető és saját összehasonlító vizsgálataink szerinti leggyorsabb módszert választottuk [1], [8].

A következőkben röviden összefoglaljuk ezt a MARSAGLIA által kidolgozott módszert [13].

Legyen $\varphi(x)$, $x \geqq 0$ az egydimenziós normális eloszlás sűrűségfüggvénye. (Az eljárás végén egy véletlen előjellel látjuk el a változót.) Konkrét alkalmazásunkban az alábbi felbontást választjuk, ahol a konstansok választását a későbbiekben magyarázzuk:

$$\varphi(x) = 0{,}9578 g_1(x) + 0{,}0395 g_2(x) + 0{,}0027 g_3(x). \tag{2.1}$$

Ily módon, ha 0,9578 valószínűséggel generálunk g_1 sűrűségfüggvényű valószínűségi változókat, 0,0395 valószínűséggel g_2 sűrűségfüggvényű és végül 0,0027 valószínűséggel g_3 sűrűségfüggvényű valószínűségi változókat, akkor az így generált valószínűségi változók összességükben φ sűrűségfüggvénnyel fognak rendelkezni.

A módszer lényeges pontja az, hogy az esetek legnagyobb részében használt g_1 sűrűségfüggvény igen egyszerű alakú, téglalapokból tevődik össze. Először a téglalapok területének megfelelő valószínűséggel kiválasztunk egy téglalapot, majd generáljuk az ennek megfelelő egyenletes eloszlású pszeudo-véletlen számot. A szükséges konstansokat egy szellemes módszer segítségével nagyon egyszerűen lehet a memóriából kikeresni — összesen 456 szóra van szükség a konstansok tárolásához. A g_1 függvény teljes leírását a [13] cikkben találhatja az olvasó. Következőkben a g_2 és a g_3 függvényt adjuk meg:

$$p_2 g_2(x) = \begin{cases} \varphi(x) - p_1 g_1(x), & 0 \leqq x \leqq 3, \\ 0, & x > 3, \end{cases}$$

$$p_3 g_3(x) = \begin{cases} \varphi(x), & x > 3, \\ 0, & 0 \leqq x \leqq 3, \end{cases}$$

ahol $p_1 = 0{,}9578$, $p_2 = 0{,}0395$ és $p_3 = 0{,}0027$.

A g_2 sűrűségfüggvényű változókat nehézkesebb előállítani, de csak az esetek 0,0395-öd részében van rá szükség, így itt egy elvetéses módszer alkalmazható; nem növeli meg lényegesen a számítógépes kiszámítás összidejét. A g_3 sűrűségfüggvényű változók fogják kiadni a normális sűrűségfüggvény végeit, ezeket egy módosított polár algoritmussal tudjuk generálni. Ez az eljárás elvileg teljesen pontos — abban az értelemben, hogy a generált változók eloszlása normális lesz —, csak a felhasznált számítógép szóhossza korlátozza a pontosságot.

A (2.1) összefüggésben szereplő konstansokat a következő módon kapjuk meg. Egy adott $g_1(x)$ függvény esetén keressük azt a legnagyobb p_1 számot, amelyre $\varphi(x) \geqq p_1 g_1(x)$, és jelen esetben $p_1 = 0{,}9578$ adódott. A $g_2(x)$ és a $g_3(x)$ függvények p_2 és p_3 szorzóit pedig úgy választottuk, hogy a p_i, $i = 1, 2, 3$ számok összege 1 legyen és a $p_3 g_3(x)$ függvény alatti terület egyenlő legyen a $\varphi(x)$, $x \geqq 3$ függvény alatti területtel.

Itt jegyezzük meg, hogy a téglalapok kiválasztását egy olyan véletlenszám generátor segítségével hajtjuk végre, amely két eltolás és egy összeadás végrehajtásával állít elő egy (0,1) intervallumban egyenletes eloszlású véletlen számot.

b) n dimenziós, **R** *korreláció mátrixú pszeudo-véletlen vektorok generálása*

Az a) pont alatti eljárás segítségével generálunk $\xi_i^{(k)}$, $i=1, 2, \ldots, n$, $k=1, \ldots, N$ pszeudo-véletlen standard normális eloszlású számokat, ezeket n dimenziós vektorok komponenseinek tekintve kapjuk a $\xi^{(k)}$, $k=1, 2, \ldots, N$ vektorokat, amelyek eloszlásfüggvénye n dimenziós normális eloszlásfüggvény lesz független komponensek esetén. Ezeket a $\xi^{(k)}$, $k=1, 2, \ldots, N$ vektorokat $\eta^{(k)}$, $k=1, 2, \ldots, N$, n dimenziós $\mathbf{R}$ korrelációs mátrixú vektorváltozókká legcélszerűbben egy $\mathbf{A}$ háromszögmátrix-szal lehet transzformálni [5], vagyis legyen

$$\eta^{(k)} = A\xi^{(k)}, \quad k = 1, 2, \ldots, N, \tag{2.2}$$

ahol az $\mathbf{A}$ mátrix a_{ij} elemét úgy határozzuk meg, hogy teljesüljön az $M(\eta_i^{(k)} \eta_j^{(k)}) = r_{ij}$ egyenlőség, ahol r_{ij} az $\mathbf{R}$ korreláció mátrix megfelelő eleme.

c) Valószínűség meghatározása beesési gyakorisággal

A fenti eljárás segítségével generált $\eta^{(k)}$, $k=1, 2, \ldots, N$ többdimenziós normális eloszlású pszeudo-véletlen vektorok segítségével torzítatlan becslést adunk a keresett $p=\Phi(\mathbf{h})$ értékre. Legyen

$$D = \{\mathbf{y} | \mathbf{y} \leqq \mathbf{h}\}, \tag{2.3}$$

és ha a generált N darab $\eta^{(k)}$ vektor közül N_1 esik a D tartományba, akkor a $P=\frac{N_1}{N}$ becslést használjuk.

A számítógépes program gyors lefutása érdekében nem számítjuk ki minden egyes k esetén a teljes $\eta^{(k)}$ vektort, hanem a következő módosítást alkalmazzuk. Egy adott $\mathbf{h}$ vektor esetén az $\eta^{(k)}$ vektor kiesik a D tartományból, ha valamilyen i index esetén $h_i < \eta_i^{(k)}$. Nyilvánvaló, hogy érdemes az $\eta^{(k)}$ vektor azon komponenseit előbb meghatározni, amelyek nagyobb valószínűséggel esnek ki a D tartományból, mint amelyek kisebb valószínűséggel esnek ki a D tartományból — hiszen ha $\eta^{(k)}$ egy komponense kiesik a D tartományból, akkor az egész $\eta^{(k)}$ vektor kiesik, és már nem kell a további komponenseket kiszámítani. Az indexeket tehát oly módon kell sorba rendezni, hogy a

$$P\{\eta_i^{(k)} > h_i\}, \quad i = 1, \ldots, n \tag{2.4}$$

valószínűségek monoton csökkenő sort alkossanak. Az $\eta_i^{(k)}$, $i=1, \ldots, n$ valószínűségi változók eloszlása az $\eta^{(k)}$ valószínűségi változók együttes eloszlásának peremeloszlása lesz, tehát egydimenziós normális eloszlás 0 várható értékkel és 1 szórással [3]. Így a h_i, $i=1, \ldots, n$ értékek növekedése alapján rendezzük sorba az indexeket, vagyis legyen

$$h_{i_1} \leqq h_{i_2} \leqq \ldots \leqq h_{i_n},$$

és ekkor az $\eta^{(k)}$ vektorok komponenseit az így kapott $i_1, i_2, \ldots, i_n$ indexsorrendben kell kiszámítani.

A módosítás előnye, hogy kis $p=\Phi(\mathbf{h})$ valószínűségek esetén csak pN esetben kell a teljes $\eta^{(k)}$ vektort kiszámítani, a többi esetekben pedig ha adott k esetén már egy j-re azt kapjuk, hogy $\eta_{i_j}^{(k)} > h_{ij}$ akkor az $\eta^{(k)}$ vektor további $\eta_{i_{j+1}}^{(k)}, \ldots, \eta_{i_n}^{(k)}$

komponenseit nem kell már kiszámítani. A fentebbi elgondolás meggyorsítja a számítógépes program lefutását, mivel a program legidőigényesebb része a (2.2) összefüggésben szereplő szorzás végrehajtása; $n=10$ dimenzióban a számítási idő 85%-át, $n=50$ dimenzióban a 96%-át köti le ez a szorzás.

3. A szubrutinok ellenőrzése és futtatási eredmények

A számítógépes program az MTA SZTAKI CDC 3300-as számítógépére készült, a szubrutinok többsége COMPASS gépi kódban — ez lényegesen hozzájárult a szubrutinok gyors lefutásához, néhány ritkábban használt szubrutin pedig FORTRAN nyelven.

A szubrutinokat négyféle módon ellenőriztük, hogy valóban a kívánt eloszlást állítják-e elő; ez egyben a felhasznált véletlenszám generátorok ellenőrzését is adta. Ellenőriztük az egydimenziós normális eloszlású számokat, a kétdimenziós és a háromdimenziós korrelált esetet és végül a független komponensű normális eloszlású $\boldsymbol{\eta}^{(k)}$ vektorok eloszlását 5—50 dimenzióig.

Mind a négyféle ellenőrzés esetén a futtatásokat a következő módon végeztük. Adott N esetén százszor futtattuk a szubrutint, a kapott valószínűségek számtani közepét jelöltük a $\overline{pr}_{comp}$ kifejezéssel (ez megfelel a $100N$ számú vektorral futtatott esetnek). A 0,95 döntési szinten kapott empirikus hibát — az egyes esetekben kapott pr_{comp} valószínűségek eltérését a $\overline{pr}_{comp}$ értéktől — h_{95}-tel jelöltük (a $\overline{pr}_{comp}-pr_{comp}$ eltérés a száz eset közül csak öt esetben nagyobb a h_{95} értéknél). A valószínűség pontos értékét pr_{exact} jelöli — ezeket a megfelelő táblázatokból kerestük, vagy számítottuk. Az adott N esetén a 0,95 valószínűségi döntési szint mellett fellépő hibára a $h=h_{95}+|pr_{exact}-\overline{pr}_{comp}|$ felső korlátot kapjuk.

Természetesen $|pr_{exact}-\overline{pr}_{comp}|\leqq\frac{1}{10}\cdot h_{95}$ 0,95 valószínűséggel. A jelöléseknél a programban található változónevekhez igazodtunk.

Minden esetben sokféle futtatást végeztünk, de a táblázatok terjedelmességének elkerülése érdekében csak néhányat közlünk; a többi eredmény a közöltekhez hasonló.

a) Az egydimenziós standard normális eloszlású számok ellenőrzése

A **h** vektor jelen esetben egydimenziós, a vizsgált értékeket $h^{(i)}$-vel jelöltük, az eredmények a következő táblázatban találhatók.

1. TÁBLÁZAT

		$N=250$			$N=2000$		
$h^{(i)}$	pr_{exact}	$\overline{pr}_{comp}$	h_{95}	h	$\overline{pr}_{comp}$	h_{95}	h
0,000	0,500 00	0,505 04	0,060	0,065	0,500 34	0,021	0,021
0,255	0,600 64	0,597 12	0,069	0,073	0,600 02	0,021	0,022
0,525	0,700 21	0,696 40	0,064	0,068	0,702 10	0,019	0,021
0,842	0,800 11	0,803 30	0,052	0,056	0,800 29	0,016	0,016
1,282	0,900 08	0,901 32	0,037	0,039	0,901 83	0,013	0,015
1,645	0,950 02	0,950 44	0,030	0,031	0,949 29	0,009	0,010
2,323	0,989 91	0,989 96	0,015	0,015	0,990 37	0,004	0,005
3,400	0,999 66	0,999 72	0,007	0,007	0,999 69	0,001	0,001

A kapott empirikus h hibák a *Csebisev egyenlőtlenség* BERNSTEJN által élesített formája szerint kapott hibáknál kisebbek [23]. Ugyanis az $N=250$ kísérlet esetén 95%-os biztonsággal adódó ε hibára a

$$P\{|\overline{pr}_{comp}-p| \geqq \varepsilon\} \leqq 2\exp\left[-\frac{N\varepsilon^2}{2pq\left(1+\frac{\varepsilon}{2pq}\right)^2}\right] \tag{3.1}$$

egyenlőtlenségből, ahol $p=\Phi(\mathbf{h})$ a keresett valószínűség és $q=1-p$, a jobboldalt egyenlővé téve 0,05-tel és a $p=q=0{,}5$ esetet véve, amelyre a legnagyobb hibát kapjuk — ε-t kifejezve az $\varepsilon \approx 0{,}090$ érték adódik, míg az empirikus h hibáinkra $h \leqq 0{,}073$ teljesül ($N=500$ esetén $\varepsilon \approx 0{,}06$ adódna a (3.1) összefüggésből).

b) Kétdimenziós korrelált eset

Az $n=2$ dimenziós normális eloszlású vektorok empirikus eloszlását az $r_1=-0{,}8$ és $r_2=+0{,}9$ korrelációs együttható és $\mathbf{h}^{(1)}=(0{,}9,\ 1{,}4)$, ill. $\mathbf{h}^{(2)}=(1{,}2,\ 1{,}8)$ határvektorok esetén ellenőriztük. A valószínűségek 10^{-6} pontosságú értékét a következő képletek alapján számítottuk ki a [24] könyvből:

$$F(h, k, \varrho) = \frac{1}{2}\Phi(h)+\frac{1}{2}\Phi(k)-T(h, a_1)-T(k, a_2)+\delta\frac{1}{2},$$

$$a_1 = \frac{k-\varrho h}{h\sqrt{1-\varrho^2}}, \qquad a_2 = \frac{h-\varrho k}{k\sqrt{1-\varrho^2}}, \tag{3.2}$$

$$\delta = \begin{cases} 1, & \text{ha} \quad hk \geqq 0, \\ 0, & \text{ha} \quad hk < 0, \end{cases}$$

ahol $F(h, k, \varrho)$ a kétdimenziós normális eloszlásfüggvény ϱ korrelációs együtthatóval a (h, k) helyen, Φ az egydimenziós normális eloszlásfüggvény, a T függvény értékei pedig a [24] könyvben találhatók. Az $r_1=-0{,}8$, $\mathbf{h}^{(1)}=(0{,}9,\ 1{,}4)$, $pr_{exact}=0{,}735189$ esetre vonatkozó eredmények a következők voltak:

2. TÁBLÁZAT

$\overline{pr}_{comp}$	N	h_{95}	h	idő
0,734 87	20 000	0,0056	0,0059	10,3 sec
0,735 47	80 000	0,0030	0,0033	41,7 sec

Az $r_2=+0{,}9$, $\mathbf{h}^{(2)}=(1{,}2,\ 1{,}8)$, $pr_{exact}=0{,}882\,637$ esetre vonatkozó eredmények pedig:

3. TÁBLÁZAT

$\overline{pr}_{comp}$	N	h_{95}	h	idő
0,882 39	20 000	0,0040	0,0042	11,6 sec
0,882 62	80 000	0,0020	0,0020	42,4 sec

A 2. és 3. táblázatban szereplő időadat (úgyanúgy, mint a továbbiakban) az adott N esetén egy lefutásnak az ideje.

c) Háromdimenziós korrelált eset

Harmadik ellenőrzési módunk az $n=3$ dimenziós eset kipróbálása volt $\mathbf{h}=(0, 0, 0)$ határvektorral és

$$\mathbf{R} = \begin{pmatrix} 1{,}0 & 0{,}8 & -0{,}4 \\ 0{,}8 & 1{,}0 & 0{,}1 \\ -0{,}4 & 0{,}1 & 1{,}0 \end{pmatrix}$$

korreláció mátrixszal. A 10^{-6} pontosságú valószínűség az

$$F_3^0 = P\{\eta_1 \leqq 0,\ \eta_2 \leqq 0,\ \eta_3 \leqq 0\} = \frac{1}{8} + \frac{1}{4\pi}[\arcsin(0{,}8) + \arcsin(-0{,}4) + \arcsin(0{,}1)]$$

képlet alapján számítható [10].

$$F_3^0 = pr_{exact} = 0{,}174\,015.$$

A számítási eredmények a 4. táblázatban találhatók.

4. TÁBLÁZAT

N	$\overline{pr}_{comp}$	h_{95}	h	idő
2 000	0,174 77	0,0182	0,0190	1,49 sec
8 000	0,174 01	0,0083	0,0083	5,58 sec
20 000	0,174 21	0,0050	0,0052	15,1 sec
50 000	0,174 18	0,0032	0,0032	33,7 sec

d) Független komponensű vektorok eloszlása 50 dimenzióig

Végül utolsó próbaként az $n=5, 10, \ldots, 50$ dimenziós független komponensű ξ vektorok eloszlását ellenőriztük. A mintaszám $N=400$ volt, ebben az esetben az empirikus hibák 0,05-nél kisebbek.

Az $n=5, 10, \ldots, 50$ esetben a $\mathbf{h}$ vektor a következő számok közül az első 5, 10, ..., 50 volt: 2,84 2,06 2,64 3,44 3,73 2,55 2,99 2,24 2,16 2,97 1,73 1,43 2,39 2,02 1,74 2,51 2,54 1,95 1,66 2,53 1,53 2,34 2,46 2,69 1,88 1,99 2,21 2,70 2,88 1,57 2,15 2,31 2,02 1,93 1,55 3,20 2,33 1,60 1,77 3,20 3,10 2,37 1,50 1,43 2,84 3,34 2,17 2,21 3,30 2,98.

A kapott eredményeket a következő táblázatban foglaltuk össze:

5. TÁBLÁZAT

n	pr_{exact}	$\overline{pr}_{comp}$	h_{95}	h	idő
5	0,973 65	0,974 07	0,016	0,017	0,99 sec
10	0,938 83	0,938 50	0,024	0,024	2,95 sec
15	0,773 02	0,773 00	0,043	0,043	5,31 sec
20	0,704 43	0,706 57	0,041	0,043	8,75 sec
25	0,627 38	0,628 57	0,046	0,047	12,1 sec
30	0,566 17	0,565 02	0,049	0,050	14,8 sec
35	0,493 20	0,494 82	0,042	0,044	20,3 sec
40	0,443 24	0,445 32	0,042	0,044	26,6 sec
45	0,377 42	0,379 17	0,046	0,048	29,7 sec
50	0,365 86	0,369 32	0,048	0,050	31,9 sec

A kapott eredmények megmutatták, hogy a szubrutinrendszer jól működik, a valószínűség értékére tetszőlegesen pontos becslést kaphatunk. Egy adott pontosság eléréséhez szükséges N mintaszámot a (3.2) egyenlőtlenség alapján határozhatunk meg.

4. A számítástechnikai tapasztalatok összefoglalása

Az egydimenziós standard normális eloszlású pszeudo-véletlen számok generálása a 2. szakasz a) pontja alatt leírt eljárás alapján igen gyorsan végezhető, másodpercenként 8000—10 000 számot lehet előállítani.

Ez a MARSAGLIA által a [13] dolgozatban publikált adatnál (10 000—12 000 szám egy másodperc alatt IBM 7090-es számítógépen) kissé rosszabb, az AHRENS—DIETER [1] szerzőpáros által megadott eredménynél (1 másodperc alatt 6000—7000 db véletlen szám IBM 360/50-es számítógépen) pedig jobb.

A többdimenziós normális eloszlás eloszlásfüggvényének kiszámítására megadott eljárásokkal hasonlítjuk össze az általunk közölt eljárást a következőkben. A MILTON által adott eljárás az általunk igényelt 5×10^{-2} pontosságot kb. 2 sec alatt éri el 5 dimenzióban (ez egy interpolált érték a [14]-ben közölt adatok alapján), míg az itt közölt eljárásnak 1 sec-re van szüksége ehhez a számításhoz. 5-nél alacsonyabb dimenziókban MILTON módszere jobb, magasabb dimenziókban az általunk közölt eljárás a gyorsabb. Lényegesen jobb a mienknél a DUTT által javasolt eljárás 4 dimenzióban [9]; 8 tizedes pontosságot a $(-\infty, +\infty)$ tartományban 0,2 sec alatt ér el. Viszont $n=4$-nél magasabb dimenzióra nem közöl eredményt. Az irodalom alapján ez tűnik a legjobb eredménynek $n=4$ és alacsonyabb dimenzióban.

A MILTON és DUTT által használt UNIVAC 1108 gép gyorsabb, mint az általunk használt CDC 3300, ennek bemutatására néhány utasítás végrehajtási idejét közöljük mikrosecundumokban:

	UNIVAC 1108	CDC 3300
Lebegőpontos összeadás	0,75	4,85—6,25
Lebegőpontos szorzás	3,00	16,00
Lebegőpontos osztás	8,75	19,00

BENE BÉLA nem végezte el az általa kidolgozott módszer részletes vizsgálatát, de módszere igen jónak tűnik: lényegében egy dimenziótól független egyenlet megoldására van csak szükség a valószínűség meghatározásához.

Az itt leírt módszer lényegesen jobb a STABIL sztohasztikus programozási modell villamosenergia-ipari alkalmazásában [21], [22] felhasznált *Monte Carlo integrálási technikánál* [7]. Összehasonlításul megemlítjük, hogy az integrálási technikával kb. 8—12 sec alatt történik egy négydimenziós valószínűség kiszámítása (0,05 hibával 95%-os biztonsági szinten), míg a dolgozatban leírt módszerrel 0,6—0,8 sec időt igényel ez a számítás.

A leírt algoritmus és számítógépes program futási ideje lényegesen függ a kiszámítandó $p=\Phi(\mathbf{h})$ valószínűség értékétől. Nagyon alacsony $p\leqq0{,}1$ és nagyon magas $p\geqq0{,}9$ valószínűségekre a (3.2) képlet miatt gyorsabb lesz a lefutás.

Másrészt mivel magasabb valószínűség esetén nagyobb futási idő szükséges a 2. c) részben leírt módosítás miatt, ezért a 3. d) részben leírt $\mathbf{h}$ vektornál nagyobb

vektorral is lefuttattuk a programot $n=50$ dimenzióban, amikor $N=400$, $\overline{pr}_{comp}=0{,}9302$, $h=0{,}022$ volt, ez 57 sec időt vett igénybe. Ebből és a (3.7) táblázatból látható, hogy a beesési gyakoriság módszerével 50 dimenziós valószínűség kiszámítása 20—40 sec alatt lehetséges. Ez megfelelő gyorsaság a STABIL modell számítógépes optimalizálási algoritmusában előforduló számítások elvégzéséhez, tehát lehetséges a STABIL modellben 50 valószínűséggel korlátozott sorral rendelkező feladat számítógépes megoldása is.

Egy-egy esetben érdekessé válhat egyetlen konkrét **h** vektor és **R** korrelációs mátrix esetén a többdimenziós normális eloszlásfüggvény kiszámítása magasabb dimenzióban. Ilyen eset lehet, ha azt vizsgáljuk, hogy egy adott lineáris programozási modell optimális megoldása mennyire valószínű, milyen valószínűséggel elégíti ki a STABIL modell alakjában felírt sztohasztizált problémát. Ezt a leírt algoritmus alapján $n=200$ dimenzió esetén is ki tudjuk számítani kb. 15 perc alatt.

IRODALOM

[1] AHRENS, J. H., DIETER, U., "Computer methods for sampling from the exponential and normal distributions", *Comm. ACM* **15** (1972) 873—882.

[2] ANDEL, J., "On multiple normal probabilities of rectangles", *Aplikace Matematiky* **16** (1971) 172—181.

[3] ANDERSON, T. W., *An introduction to the multivariate statistical analysis* (J. Wiley, New York, 1958).

[4] BENE, B., »Reichentransformationen zur gemeinsamen Normalverteilung von familiären Erkrankungsneigungen«, in: *Proceedings of the Symposium on Computational Statistics* (Wien, 1974).

[5] BUSZLENKO, N. P., GOLENKO, D. I., SREJGYER, JU. A., SZOBOL, I. M., SZRAGOVICS, V. G., *Monte Carlo módszerek* (Műszaki Könyvkiadó, Budapest, 1965).

[6] DEÁK, I., „Egy sztohasztikus programozási modell számítógépes kiértékelése", *MTA Számítástechnikai Központ Közleményei* **9** (1972) 33—49.

[7] DEÁK, I., „A többdimenziós normális eloszlásfüggvény Monte Carlo integrálással történő kiszámításának számítógépes tapasztalatai", *MTA Számítástechnikai és Automatizálási Intézet Közleményei* **18** (1977) 5—17.

[8] DIETER, U., AHRENS, J. H., "A combinatorial method for the generation of normally distributed random numbers", *Computing* **11** (1973) 137—146.

[9] DUTT, J. E., "A representation of multivariate normal probability integrals by integral transforms", *Biometrika* **60** (1973) 637—645.

[10] GUPTA, S. S., "Probability integrals of multivariate normal and multivariate t", *Ann. Math. Stat.* **34** (1963) 792—829.

[11] JOHNSON, N. L., KOTZ, S., *Distributions in Statistics IV. vol.* (J. Wiley, New York 1972).

[12] MARSAGLIA, G., "One-sided approximations by linear combinations of functions", in: *Proceedings of Symposium on Approximation Theory* (held at Lancaster 1969, Academic press, New York 1970) 233—242.

[13] MARSAGLIA, G., MACLAREN, M. D., BRAY, T. A., "A fast procedure for generating normal random variables", *Comm. ACM* **7** (1964) 4—10.

[14] MILTON, R. C., "Computer evaluation of the multivariate normal integral", *Technometrics* **14** (1972) 881—889.

[15] PRÉKOPA, A., "Stochastic programming models for inventory control and water storage problems", in: *Inventory Control and Water Storage*, Ed. A. Prékopa (János Bolyai Mathematical Society and North-Holland Publishing Company, Amsterdam—London, 1973) 229—245.

[16] PRÉKOPA, A., "On probabilistic constrained programming", in: *Proceedings of the Princeton Symposium on Mathematical Programming* (Princeton University Press, Princeton, N. J., 1970) 113—138.

[17] PRÉKOPA, A., „Sztochasztikus rendszerek optimalizálási problémáiról" doktori értekezés, Magyar Tudományos Akadémia, Budapest, 1970.

[18] PRÉKOPA, A., "Logaritmic concave measures with applications to stochastic programming", *Acta Scientiarum Mathematicarum* **32** (1971) 301—316.

[19] PRÉKOPA, A., „A *megengedett irányok* elnevezésű nemlineáris programozási módszer kiterjesztése kvázikonkáv feltételi függvények esetére", *MTA Számítástechnikai Központ Közleményei* **9** (1972) 3—16.
[20] PRÉKOPA, A., "Contributions to the theory of stochastic programming", *Mathematical Programming* **4** (1973) 202—221.
[21] PRÉKOPA, A., DEÁK, I., GANCZER, S. és PATYI, K., „A STABIL sztochasztikus programozási modell és annak kísérleti alkalmazása a magyar villamosenergia-iparra", *Alkalmazott Matematikai Lapok* **1** (1975) 3—22.
[22] PRÉKOPA, A., DEÁK, I., GANCZER, S. and PATYI, K., "The STABIL stochastic programming model and its experimental application to the electrical energy sector of the Hungarian economy", in: *Proceedings of the Symposium on Stochastic Programming* (Oxford 1975) (sajtó alatt).
[23] RÉNYI, A., *Valószínűségszámítás* (Tankönyvkiadó, Budapest, 1966).
[24] СМИРНОВ, Н. В. и БОЛЬШЕВ, Л. Н., *Таблицы для вычисления функции двухмерного нормального распределения* (Изд. Ак. Наук, Москва, 1962).

DEÁK ISTVÁN
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1502 BUDAPEST XI., KENDE ÚT 15—17.

ON MULTIPLE NORMAL PROBABILITIES OF SPECIAL SETS

I. DEÁK

A computer algorithm is presented which computes the numerical values of probabilities connected with multidimensional normal distribution. The subroutines working on the base of the described technique compute the concrete values of the distribution function of the multidimensional normal distribution with arbitrary correlation matrix up to 50 dimensions; the errors and the times are adequate in practice. The results of the checking computions and the times of the runs are also presented.

Alkalmazott Matematikai Lapok **2** *(1976) 27—39*

EGY ELJÁRÁS A TÖBBDIMENZIÓS NORMÁLIS ELOSZLÁSFÜGGVÉNY ÉS GRADIENSE ÉRTÉKEINEK MEGHATÁROZÁSÁRA

SZÁNTAI TAMÁS
Budapest

A dolgozatban leírunk egy algoritmust a többdimenziós normális eloszlásfüggvény értékének a számítására, és megmutatjuk, hogy hogyan használható az egyúttal a gradiens vektor számolására is. Az algoritmus a sztochasztikus programozási feladatok megoldása szempontjából különösen érdekes, egyhez közeli valószínűség értékek számítására igen hatékony. Közöljük az agloritmus FORTRAN nyelven megírt programját és a próbafuttatások során nyert eredményeket.

1. Bevezetés

Legyenek ξ_i, $i=1, 2, \ldots, n$ nulla várható értékű, egy szórású és $\mathbf{R}$ korreláció mátrixú, normális együttes eloszlású valószínűségi változók. Az

$$P(\xi_i \leqq x_i, \quad i = 1, 2, \ldots, n) \tag{1.1}$$

valószínűség értékét, valamint az

$$\frac{\partial P(\xi_i \leqq x_i,\ i=1, 2, \ldots, n)}{\partial x_l}, \quad l = 1, 2, \ldots, n \tag{1.2}$$

parciális deriváltak értékeit kívánjuk meghatározni, ahol x_i, $i=1, 2, \ldots, n$ tetszőleges valós számok.

Tekintettel arra, hogy minden nagyobb számítógépen könyvtári programként rendelkezésre áll egy független, standard normális eloszlású véletlen számokat generáló szubrutin, az (1.1) érték közelítő meghatározására egy viszonylag egyszerű, de meglehetősen munkaigényes eljárás a következő.

Jelölje $\varepsilon_i^{(s)}$, $i=1, 2, \ldots, n$, $s=1, 2, \ldots, S$ a független, standard normális eloszlású véletlen szám generáló szubrutin által előállított számsorozatot. Mivel az $\mathbf{R}$ korreláció mátrix pozitív definit és szimmetrikus, létezik és viszonylag egyszerűen számolható egy olyan $\mathbf{A}$ felső háromszög alakú mátrix, amelyre

$$\mathbf{A}\mathbf{A}' = \mathbf{R}. \tag{1.3}$$

Jelölje az **A** mátrix nem nulla elemeit a_{ik}, $i=1, 2, \ldots, n$, $k=i, i+1, \ldots, n$ és képezzük a következő véletlen számsorozatot:

$$\begin{aligned} \eta_1^{(s)} &= a_{11}\varepsilon_1^{(s)} + a_{12}\varepsilon_2^{(s)} + \ldots + a_{1n}\varepsilon_n^{(s)}, \\ \eta_2^{(s)} &= \quad a_{22}\varepsilon_2^{(s)} + \ldots + a_{2n}\varepsilon_n^{(s)}, \\ &\vdots \qquad\qquad \ddots \quad \vdots \\ \eta_n^{(s)} &= \qquad\qquad a_{nn}\varepsilon_n^{(s)}, \\ & s = 1, 2, \ldots, S. \end{aligned} \tag{1.4}$$

Az így előállított $\eta_i^{(s)}$, $i = 1, 2, \ldots, n$, $s = 1, 2, \ldots, S$ véletlen számok a ξ_i, $i=1, 2, \ldots, n$ valószínűségi változók egy S elemű mintáját alkotják. Ezért, ha összeszámoljuk, hogy hány s értékre teljesül az

$$\eta_i^{(s)} \leqq x_i, \quad i = 1, 2, \ldots, n \tag{1.5}$$

feltételek mindegyike egyidejűleg, és az így nyert számot S-sel elosztjuk, az (1.1) valószínűség értékre egy relatív gyakoriság értékét nyerünk, és ez elég nagy S mintaszám mellett a tényleges valószínűség értékét elfogadható mértékben megközelíti. $S=400$ esetén például elég kicsi, illetve elég nagy valószínűség értékek számításakor a közelítés hibája nagy megbízhatósággal 0,05 körüli értékű (lásd [1]). Ez a pontosság azonban nem mindig elegendő egy olyan sztochasztikus programozási feladat megoldása során, amikor az előírt valószínűségi szint 0,95, vagy annál is nagyobb.

Az (1.2) parciális deriváltak meghatározására a [3] dolgozatban a szerzők a következő eljárást használják. Jelölje $\Phi(x_1, x_2, \ldots, x_n; \mathbf{R})$ az (1.1) valószínűség értékét, mint az $x_1, x_2, \ldots, x_n$ változók, valamint az **R** korreláció mátrix függvényét. A valószínűségelméletből ismert, hogy a $\Phi(x_1, x_2, \ldots, x_n; \mathbf{R})$ függvény x_l szerinti parciális deriváltja és a $\xi_1, \xi_2, \ldots, \xi_{l-1}, \xi_{l+1}, \ldots, \xi_n$ valószínűségi változók $\Phi(x_1, x_2, \ldots, x_{l-1}, x_{l+1}, \ldots, x_n|x_l)$ szimbólummal jelölt, $\xi_l = x_l$ feltétel melletti feltételes eloszlásfüggvénye között fennáll az

$$\frac{\partial \Phi(x_1, x_2, \ldots, x_n; \mathbf{R})}{\partial x_l} = \Phi(x_1, x_2, \ldots, x_{l-1}, x_{l+1}, \ldots, x_n|x_l)\varphi(x_l) \tag{1.6}$$

kapcsolat, ahol φ az egyváltozós standard normális eloszlás sűrűségfüggvénye. Ismeretes továbbá, hogy

$$\Phi(x_1, x_2, \ldots, x_{l-1}, x_{l+1}, \ldots, x_n|x_l) = \tag{1.7}$$

$$= \Phi\left(\frac{x_1 - r_{1l}x_l}{\sqrt{1-r_{1l}^2}}, \frac{x_2 - r_{2l}x_l}{\sqrt{1-r_{2l}^2}}, \ldots, \frac{x_{l-1} - r_{l-1,l}x_l}{\sqrt{1-r_{l-1,l}^2}}, \frac{x_{l+1} - r_{l,l+1}x_l}{\sqrt{1-r_{l,l+1}^2}}, \ldots, \frac{x_n - r_{ln}x_l}{\sqrt{1-r_{ln}^2}}; \mathbf{R}^{(l)}\right),$$

ahol az $\mathbf{R}^{(l)}$ $(n-1)\times(n-1)$ méretű korreláció mátrix a következő elemekből áll:

$$r_{ik}^{(l)} = \frac{r_{ik} - r_{il}r_{kl}}{\sqrt{1-r_{il}^2}\sqrt{1-r_{kl}^2}}, \quad i, k = 1, 2, \ldots, n; \; i \neq l, \; k \neq l. \tag{1.8}$$

Az (1.6) és (1.7) összefüggésekből azt kapjuk, hogy

$$\frac{\partial \Phi(x_1, x_2, \ldots, x_n; \mathbf{R})}{\partial x_l} = \tag{1.9}$$

$$= \Phi\left(\frac{x_1 - r_{l1}x_l}{\sqrt{1-r_{l1}^2}}, \frac{x_2 - r_{l2}x_l}{\sqrt{1-r_{l2}^2}}, \ldots, \frac{x_{l-1} - r_{l,l-1}x_l}{\sqrt{1-r_{l,l-1}^2}}, \right.$$

$$\left. \frac{x_{l+1}r_{-l,l+1}x_l}{\sqrt{1-r_{l,l+1}^2}}, \ldots, \frac{x_n - r_{ln}x_l}{\sqrt{1-r_{ln}^2}}; \mathbf{R}^{(l)}\right) \varphi(x_l),$$

vagyis az n-dimenziós normális eloszlásfüggvény parciális deriváltjainak a számolása visszavezethető $(n-1)$ dimenziós normális eloszlásfüggvény értékek számolására, ahol az $(n-1)$-dimenziós $\mathbf{R}^{(l)}$ korreláció mátrixok az (1.8) alatt definiált elemekből állnak, $l=1, 2, \ldots, n$.

A következő szakaszban leírunk egy algoritmust a többdimenziós normális eloszlásfüggvény értékének a számolására, amely végrehajtása a fentebb leírt egyszerű eljárás végrehajtásával azonos nagyságrendű időt igényel. A harmadik szakaszban közöljük mind az eloszlásfüggvényérték, mind a gradiens vektor számítására megírt FORTRAN nyelvű programokat. Végül a negyedik szakaszban néhány teszt számítás eredményein illusztráljuk az algoritmus hatékonyságát.

2. Az algoritmus leírása

Az algoritmus alapja az (1.1) valószínűségnek az *általános valószínűségi* (vagy általánosabban *Jordan*) *tétel* (lásd pl. [2]) segítségével történő, következő átalakítása:

$$P(\xi_i \leqq x_i,\ i = 1, 2, \ldots, n) = 1 - \sum_{i=1}^{n} P(\xi_i \geqq x_i) + \tag{2.1}$$

$$+ \sum_{1 \leqq i < j \leqq n} P(\xi_i \geqq x_i,\ \xi_j \geqq x_j) - \sum_{1 \leqq i < j < k \leqq n} P(\xi_i \geqq x_i, \xi_j \geqq x_j,\ \xi_k \geqq x_k) +$$

$$+ \ldots + (-1)^n P(\xi_i \geqq x_i,\ i = 1, 2, \ldots, n) = 1 - n + \sum_{i=1}^{n} \Phi(x_i) +$$

$$+ \sum_{1 \leqq i < j \leqq n} P(\xi_i \geqq x_i,\ \xi_j \geqq x_j) - \sum_{1 \leqq i < j < k \leqq n} P(\xi_i \geqq x_i,\ \xi_j \geqq x_j,\ \xi_k \geqq x_k) +$$

$$+ \ldots + (-1)^n P(\xi_i \geqq x_i,\ i = 1,2, \ldots, n).$$

Ha bevezetjük a

$$D(x_1, x_2, \ldots, x_n) = 1 - n + \sum_{i=1}^{n} \Phi(x_i), \tag{2.2}$$

valamint a

$$H(x_1, x_2, \ldots, x_n) = \sum_{1 \leqq i < j \leqq n} P(\xi_i \geqq \xi x_{i,j} \geqq x_j) - \sum_{1 \leqq i < j < k \leqq n} P(\xi_i \geqq x_i, \tag{2.3}$$

$$\xi_j \geqq x_j,\ \xi_k \geqq x_k) + \ldots + (-1)^n P(\xi_i \geqq x_i,\ i = 1, 2, \ldots, n)$$

jelöléseket, ahol $\Phi(x)$ az egydimenziós standard normális eloszlás eloszlásfüggvényét jelöli, akkor a (2.1) összefüggés a következő alakba írható:

$$P(\xi_i \leqq x_i,\ i = 1, 2, \ldots, n) = D(x_1, x_2, \ldots, x_n) + H(x_1, x_2, \ldots, x_n). \tag{2.4}$$

Annak ellenére, hogy a $D(x_1, x_2, \ldots, x_n)$ kifejezés értékét igen gyorsan és pontosan tudjuk számítani, az első pillantásra nem látszik túl célszerűnek a $P(\xi_i \leqq x_i,\ i=1, 2, \ldots, n)$ valószínűség fenti átalakítása, hiszen a $H(x_1, x_2, \ldots, x_n)$ kifejezés értékének a számolása csak szimulációs módszerrel lehetséges, mely során egy helyett 2^n-n-1 számú valószínűségérték számítását kell végrehajtani. Meg fogjuk mutatni azonban, hogy a $H(x_1, x_2, \ldots, x_n)$ kifejezésbe foglalt 2^n-n-1 számú valószínűség együttes értékének szimulációs módszerrel történő meghatározása csak látszólag nehezebb feladat, mint az egyetlen $P(\xi_i \leqq x_i,\ i=1, 2, \ldots, n)$ valószínűség értékének a számítása. Ugyanakkor egyhez közeli értékű valószínűségek számolásakor a legtöbb esetben a gyorsan és pontosan számítható $D(x_1, x_2, \ldots, x_n)$ kifejezés dominálja a teljes $P(\xi_i \leqq x_i,\ i=1, 2, \ldots, n)$ valószínűség értéket.

A $H(x_1, x_2, \ldots, x_n)$ kifejezésben szereplő

$$\sum_{1 \leqq i < j \leqq n} P(\xi_i \geqq x_i, \xi_j \geqq x_j) - \sum_{1 \leqq i < j < k \leqq n} P(\xi_i \geqq x_i, \xi_j \geqq x_j, \xi_k \geqq x_k) +$$

$$+ \ldots + (-1)^n P(\xi_i \geqq x_i,\ i = 1, 2, \ldots, n)$$

valószínűségek együttes értékének szimulációs módszerrel történő közelítésekor ismét az első szakaszban definiált $\eta_i^{(s)}$, $i=1, 2, \ldots, n$, $s=1, 2, \ldots, S$ véletlen számokat használhatjuk. A feladatunk most az, hogy minden s értékre összeszámoljuk, hogy a

$$\begin{aligned}
&F_{ij}^{(s)}: \eta_i^{(s)} \geqq x_i,\ \eta_j^{(s)} \geqq x_j, \quad 1 \leqq i \leqq n, \\
&F_{ijk}^{(s)}: \eta_i^{(s)} \geqq x_i,\ \eta_j^{(s)} \geqq x_j,\ \eta_k^{(s)} \geqq x_k, \quad 1 \leqq i < j < k \leqq n, \\
&\qquad\vdots \\
&F_{12\ldots n}^{(s)}: \eta_1^{(s)} \geqq x_1,\ \eta_2^{(s)} \geqq x_2, \ldots, \eta_n^{(s)} \geqq x_n
\end{aligned} \tag{2.5}$$

feltételek közül mennyivel több páros számú alsó indexszel rendelkező teljesül, mint ahány páratlan számú alsó indexszel rendelkező. Az így nyert számokat $s=1, 2, \ldots, S$-re összegezve és S-sel elosztva, egy a $H(x_1, x_2, \ldots, x_n)$ kifejezés értékét közelítő relatív gyakoriságot nyerünk.

A (2.5) feltételek rögzített s érték melletti teljesülésének az ellenőrzését úgy végezhetjük, hogy sorra ellenőrizzük az $\eta_i^{(s)} \geqq x_i$, $i=1, 2, \ldots, n$ feltételek teljesülését, és amikor egy éppen teljesülő feltételhez jutunk, összeszámoljuk, hogy az ezáltal minden részfeltételében teljesülővé váló F feltételek között mennyivel több a páros számú alsó indexű, mint a páratlan számú alsó indexű, és a továbbiakban ezeket az F feltételeket figyelmen kívül hagyjuk. Az így nyert számokat $i=1, 2, \ldots, n$-re összegezve a kívánt értéket nyerjük. A számolást tovább egyszerűsíti az az észrevétel, hogy egy $\eta_j^{(s)} \geqq x_j$ feltétel teljesülésének a hatására függetlenül attól, hogy előzőleg hány (de legalább egy) és mely $\eta_i^{(s)} \geqq x_i$, $i<j$ feltételek teljesültek, mindig pontosan eggyel több páros alsó indexű F feltétel teljesülése következik be újonnan, mint ahány páratlan alsó indexűé. Jelölje ugyanis $k(\geqq 1)$ a j-edik $\eta_j^{(s)} \geqq x_j$ feltétel vizsgálata előtt már teljesült $\eta_i^{(s)} \geqq x_i$ feltételek számát. Ekkor az újonnan teljesülő

F feltételek úgy írhatók le, mint az $\eta_j^{(s)} \geqq x_j$ feltétel mellé a már teljesültek közül összes lehetséges módon kiválasztott $1, 2, \ldots, k$ számú feltételek által alkotott feltétel halmazok. Ezek megfelelő előjellel ellátott összege:

$$\binom{k}{1} - \binom{k}{2} + \binom{k}{3} - \ldots + (-1)^{k-1}\binom{k}{k} = \binom{k}{0} = 1.$$

Mivel a fenti módon eljárva az $\eta_i^{(s)} \geqq x_i$, $i=1, 2, \ldots, n$ feltételek mindegyikének a teljesülését pontosan csak egyszer kellett ellenőriznünk, és a teljesüléstől függően egy összeadást végrehajtani, vagy nem, a számolási idő valóban azonos nagyságrendű az (1.5) feltételek teljesülésének az ellenőrzéséhez szükséges idővel. (Az utóbbi egyetlen előnnyel bír, nevezetesen, hogy ha valamely feltétel nem teljesül, az összes többi vizsgálata elhagyható. Ez azonban egyhez közeli valószínűségértékek számolása esetén csak ritkán következik be.)

Könnyen látható, hogy ha az egyes feltételek teljesülésétől függően egy-egy újabb összeadást hajtunk végre, vagy nem hajtunk végre, akkor egyszerre nyerhetjük a $P(\xi_i \leqq x_i, i=1, 2, \ldots, n)$ valószínűség és a $H(x_1, x_2, \ldots, x_n)$ kifejezés szimulációs közelítésének az értékét. Jelölje PL az előbbi és PH az utóbbi közelítés értékét. Ekkor PL és $1-n+\sum_{i=1}^{n} \Phi(x_i)+PH$ az (1.1) valószínűség két, egymástól különböző becslését jelenti. Ezeknek egy megfelelően súlyozott átlaga várhatóan pontosabb közelítést eredményez, mint bármelyikük külön-külön. A súlyokat célszerű a PL és a PH becslések becsült szórásával fordított arányban választani. Ha a becsléseket S elemszámú véletlen mintára támaszkodva nyertük, akkor PL becsült szórása:

$$\sqrt{\frac{PL(1-PL)}{S}}, \tag{2.6}$$

PH becsült szórásának pedig a

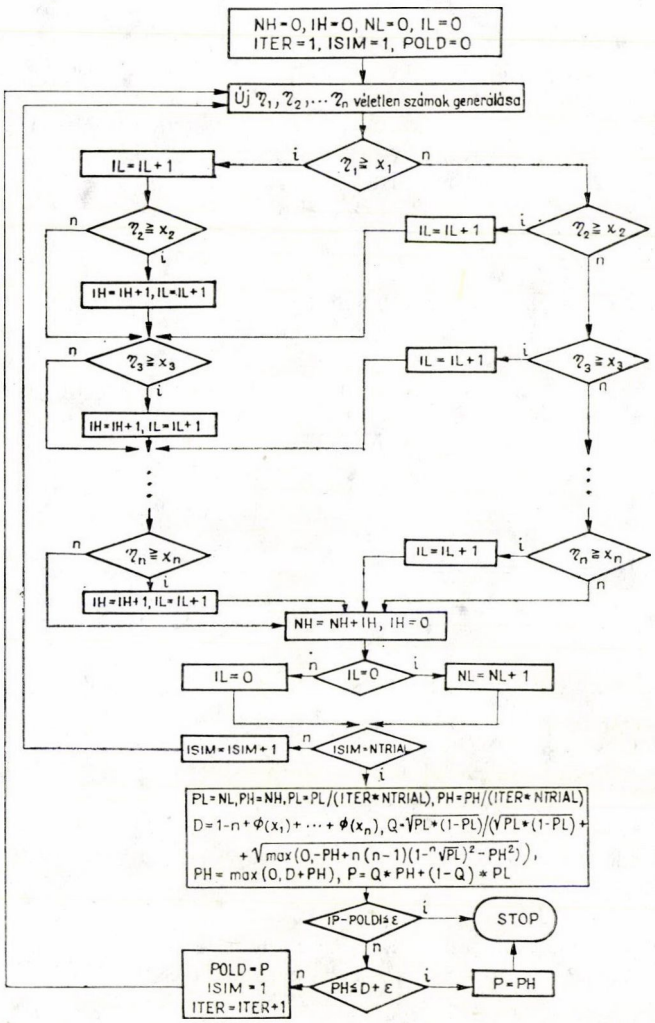
$$\sqrt{\frac{\max\left(0, -PH+n(n-1)\left(1-\sqrt[n]{PL}\right)^2-PH^2\right)}{S}} \tag{2.7}$$

értéket fogadjuk el. A (2.6) érték PL szórásának torzítatlan becslése, a (2.7) érték pedig PH szórásának olyan becslése, amely csak a független és azonos argumentumú $(x_1=x_2=\ldots=x_n)$ esetben torzítatlan.[1] A (2.6) és (2.7) értékekkel fordított arányban elkészített, SL és SH-val jelölt súlyokat különböző n és P értékekre az 1. táb-

[1] Ekkor ugyanis PH olyan v_s, $s=1, 2, \ldots, S$ független valószínűségi változók átlaga, amelyek a $0, 1, 2, \ldots, n-1$ értékeket $p_0=(1-p)^n+\binom{n}{1}p(1-p)^{n-1}$ és $p_i=\binom{n}{i+1}p^{i+1}(1-p)^{n-i-1}$, $i=$ $=1, 2, \ldots, n-1$ valószínűségekkel veszik fel. Így a várható értéke $(1-p)^n-1+np$, vagy a $p=1-\sqrt[n]{P}$ jelölés bevezetésével a $P-1+n\left(1-\sqrt[n]{P}\right)$, a szórásnégyzete pedig $-(1-p)^n+1-np+n(n-1)p^2-$ $-[(1-p)^n-1+np]^2$, vagy P-vel kifejezve: $-P+1-n\left(1-\sqrt[n]{P}\right)+n(n-1)\left(1-\sqrt[n]{P}\right)^2-[P-1+$ $+n\left(1-\sqrt[n]{P}\right)]^2$.

lázat tartalmazza. Ebből leolvasható, hogy 0,4 körüli és annál nagyobb P valószínűség számolásakor az $1-n+\sum_{i=1}^{n}\Phi(x_i)+PH$ közelítő értéket nagyobb súllyal vesszük figyelembe, mint a PL közelítő értéket. Megjegyezzük, hogy PH becsült szórását érdemesebb lehet a rendelkezésre álló véletlen mintából számítani, semmint a gyakran pontatlan (2.7) közelítést alkalmazni.

A leírt algoritmust valósítja meg az 1. ábra blokkdiagramja úgy, hogy az előírt pontosság elérése érdekében szükség esetén egyre több és több véletlen számra támaszkodva hajtja végre a szimulációs kiértékelést.



1. ábra

1. TÁBLÁZAT. Az *SL* és az *SH* súlyok különböző *n* és *P* értékekre

n / *P*	0.1		0.2		0.3		0.4		0.5		0.6		0.7		0.8		0.9	
	SL	*SH*	*SL*	*SH*	*SL*	*SH*	*SL*	*SH*	*SL*	*SH*	*SL*	*SH*	*SL*	*SH*	*SL*	*SH*	*SL*	*SH*
2	0,625	0,375	0,535	0,465	0,468	0,532	0,411	0,589	0,359	0,641	0,310	0,690	0,260	0,740	0,208	0,792	0,146	0,854
3	0,705	0,295	0,610	0,390	0,535	0,465	0,471	0,529	0,412	0,588	0,355	0,645	0,298	0,702	0,238	0,762	0,167	0,833
4	0,739	0,261	0,642	0,358	0,565	0,435	0,498	0,502	0,435	0,565	0,375	0,625	0,315	0,685	0,251	0,749	0,176	0,824
5	0,758	0,242	0,661	0,339	0,583	0,417	0,513	0,487	0,449	0,551	0,387	0,613	0,325	0,675	0,259	0,741	0,181	0,819
6	0,770	0,230	0,673	0,327	0,594	0,406	0,523	0,477	0,458	0,542	0,395	0,605	0,331	0,669	0,264	0,736	0,185	0,815
7	0,778	0,222	0,681	0,319	0,601	0,399	0,530	0,470	0,464	0,536	0,400	0,600	0,336	0,664	0,267	0,733	0,187	0,813
8	0,784	0,216	0,687	0,313	0,607	0,393	0,536	0,464	0,469	0,531	0,404	0,596	0,339	0,661	0,270	0,730	0,189	0,811
9	0,789	0,211	0,691	0,309	0,611	0,389	0,540	0,460	0,472	0,528	0,407	0,593	0,342	0,658	0,272	0,728	0,190	0,810
10	0,792	0,208	0,695	0,305	0,615	0,385	0,543	0,457	0,475	0,525	0,410	0,590	0,344	0,656	0,274	0,726	0,192	0,808
11	0,795	0,205	0,698	0,302	0,618	0,382	0,545	0,455	0,478	0,522	0,412	0,588	0,345	0,655	0,275	0,725	0,193	0,807
12	0,798	0,202	0,700	0,300	0,620	0,380	0,548	0,452	0,480	0,520	0,413	0,587	0,347	0,653	0,276	0,724	0,193	0,807
13	0,800	0,200	0,702	0,298	0,622	0,378	0,549	0,451	0,481	0,519	0,415	0,585	0,348	0,652	0,277	0,723	0,194	0,806
14	0,801	0,199	0,704	0,296	0,624	0,376	0,551	0,449	0,483	0,517	0,416	0,584	0,349	0,651	0,278	0,722	0,194	0,806
15	0,803	0,197	0,706	0,294	0,625	0,375	0,552	0,448	0,484	0,516	0,417	0,583	0,350	0,650	0,278	0,722	0,195	0,805
16	0,804	0,196	0,707	0,293	0,626	0,374	0,553	0,447	0,485	0,515	0,418	0,582	0,350	0,650	0,279	0,721	0,195	0,805
17	0,805	0,195	0,708	0,292	0,627	0,373	0,554	0,446	0,486	0,514	0,419	0,581	0,351	0,649	0,280	0,720	0,196	0,804
18	0,806	0,194	0,709	0,291	0,628	0,372	0,555	0,445	0,486	0,514	0,419	0,581	0,352	0,648	0,280	0,720	0,196	0,804
19	0,807	0,193	0,710	0,290	0,629	0,371	0,556	0,444	0,487	0,513	0,420	0,580	0,352	0,648	0,280	0,720	0,196	0,804
20	0,808	0,192	0,711	0,289	0,630	0,370	0,557	0,443	0,488	0,512	0,421	0,579	0,353	0,647	0,281	0,719	0,196	0,804

A parciális deriváltak számításáról megjegyezzük, hogy az (1.9) képletben szereplő $(n-1)$-dimenziós normális eloszlásfüggvény értéket a (2.1) összefüggés szerint átalakítva, a szimulációval meghatározandó tagok azonosak lesznek a (2.1) összefüggés jobb oldalán szereplő három és annál több ξ_i valószínűségi változóra vonatkozó tagok parciális deriváltjaival. Minthogy ezek a jobboldali tagok általában csak igen kicsiny hányadát képviselik a teljes (1.1) valószínűségértéknek, várható, hogy a parciális deriváltak értékében sem fognak jelentős szerepet játszani.

3. A FORTRAN programok listája

Az (1.1) valószínűséget számító program függvény eljárás alakú. A P= =PLNORM (R, X, N, NMAX, NTRIAL, EPS, IRAND) utasítás hatására P értéke az R korreláció mátrixú, standard N-dimenziós normális eloszlásfüggvény X(1), X(2), ..., X(N) helyen felvett értékével lesz egyenlő. A paraméterek jelentése:

R — kétdimenziós tömb a korreláció mátrix inputjára,
X — egydimenziós tömb az eloszlásfüggvény argumentumainak az inputjára,
N — a normális eloszlás dimenziószáma,
NMAX — az R és az X tömböknek a hívó programban deklarált mérete,
NTRIAL — a szimulációs kiértékelés során egyszerre vizsgálandó véletlen vektorok száma (az értéke lehet például 100, 250 vagy 500 a kívánt pontosság és gyorsaság szerint),
EPS — a számolás leállításának a kritériuma, ha az utolsónak vizsgált véletlen számsorozat az eloszlásfüggvény közelítő értékét ennél a számnál kisebb mértékben módosítja, a szubrutin befejezi a számolást,
IRAND — tetszőleges egész szám, mely a véletlen szám generáló rutin induló értékéül szolgál.

A program két szubrutint használ, melyek közül a SUPNOR(BUFF, NN, IRAND) szubrutin a BUFF nevű, NN méretű, egydimenziós valós tömböt tölti fel független, standard normális eloszlású véletlen számokkal. Az IRAND paraméter itt is a véletlen szám generálás kiinduló pontjául szolgál. Az RNORM(X) függvényeljárás az egydimenziós standard normális eloszlásfüggvény értékét számítja. Mindkét program megtalálható a nagyobb számítógépek programkönyvtárában.

A gradiens vektort számító program szubrutin alakú. A

CALL GRNORM(R, X, GRAD, N, NMAX, NTRIAL, EPS, IRAND)

utasítás hatására a GRAD(1), GRAD(2), ..., GRAD(N) tömbelemek értékei az R korreláció mátrixú, standard N-dimenziós normális eloszlásfüggvény X(1), X(2), ..., X(N) helyen számított gradiens vektorának a komponenseivel lesznek egyenlők. A paraméterek jelentése azonos a PLNORM függvényeljárás paramétereinek a jelentésével, eltekintve a GRAD paramétertől, amely a gradiens vektor outputjául szolgál. A program használja a PLNORM függvényeljárást, és ezen keresztül a már ismertetett SUPNOR és RNORM programokat.

A programok listája a következő:

```
      FUNCTION PLNORM(R, X, N, NMAX, NTRIAL, EPS, IRAND)
      DIMENSION R(NMAX, NMAX), X(NMAX), A(20, 20), ETA(20),
     1 BUFF(2000)
```

```
C***A TRANSZFORMACIOS MATRIX SZAMITASA.
      DO 1 I=1, N
      DO 1 J=1, N
    1 A(I, J)=0.0
      DO 2 I=1, N
    2 A(I, N)=R(I, N)
      DO 3 JJ=2, N
      J=N-JJ+1
      A(J, J)=R(J, J)
      JP1=J+1
      DO 4 L=JP1, N
    4 A(J, J)=A(J, J)-A(J, L)*A(J, L)
      A(J, J)=SQRT(ABS(A(J, J)))
      JM1=J-1
      IF(JM1.EQ.0) GO TO 3
      DO 5 I=1, JM1
      A(I, J)=R(I, J)
      DO 6 L=JP1, N
    6 A(I, J)=A(I, J)-A(I, L)*A(J, L)
    5 A(I, J)=A(I, J)/A(J, J)
    3 CONTINUE
C***A RELATIV GYAKORISAGOK SZAMITASA.
      NH=0
      IH=0
      NL=0
      IL=0
      POLD=0.0
      ITER=1
      ISIM=1
C***AZ UJ VELETLEN SZAMOK GENERALASA.
   21 CALL SUPNOR(BUFF, NTRIAL*N, IRAND)
   22 N1=N*(ISIM-1)
      DO 23 I=1, N
      ETA(I)=0.0
      DO 23 J=I, N
      K=N1+J
   23 ETA(I)=ETA(I)+A(I, J)*BUFF(K)
C***A FELTETELEK SOROZATANAK A VIZSGALATA.
      IF(ETA(1).LT.X(1)) GO TO 24
      IL=IL+1
      I=2
      GO TO 25
   24 I=2
   26 IF(ETA(I).LT.X(I)) GO TO 27
      IL=IL+1
      GO TO 28
   27 IF(I.EQ.N) GO TO 30
      I=I+1
```

```
      GO TO 26
   25 IF(ETA(I).LT.X(I)) GO TO 28
      IH=IH+1
      IL=IL+1
   28 IF(I.EQ.N) GO TO 30
      I=I+1
      GO TO 25
   30 NH=NH+IH
      IH=0
      IF(IL.EQ.0) GO TO 31
      IL=0
      GO TO 32
   31 NL=NL+1
   32 IF(ISIM.EQ.NTRIAL) GO TO 41
      ISIM=ISIM+1
      GO TO 22
C*** A VALOSZINUSEG ERTEK SZAMITASA.
   41 PL=NL
      PH=NH
      PL=PL/(ITER*NTRIAL)
      PH=PH/(ITER*NTRIAL)
      Q=SQRT(PL*(1.-PL))/(SQRT(PL*(1.-PL))
     1  +SQRT(AMAX1(0.,-PH-PH**2+N*(N-1)*(1.-PL**
      (1./N))**2)))
      D=1.-N
      DO 42 I=1, N
   42 D=D+RNORM(X(I))
      PH=AMAX1(0., D+PH)
      P=Q*PH+(1.-Q)*PL
      IF(ABS(P-POLD).LT.EPS) GO TO 43
      IF(PH.LT.D+EPS) GO TO 44
      POLD=P
      ISIM=1
      ITER=ITER+1
      GO TO 21
   43 PLNORM=P
      GO TO 50
   44 PLNORM=PH
   50 RETURN
      END
      SUBROUTINE GNORM(R, X, GRAD, N, NMAX, NTRIAL, EPS, IRAND)
      DIMENSION R(NMAX, NMAX), X(NMAX), GRAD(NMAX),
      1RNEW(20, 20), XNEW(20)
      DO 1 L=1, N
      IS=0
      DO 11 I=1, N
      IF(I.EQ.L) GO TO 11
      IS=IS+1
```

```
      XNEW(IS)=(X(I)-R(L, I)*X(L))/SQRT(1.-R(L, I)**2)
   11 CONTINUE
      IS=0
      DO 13 I=1, N
      IF(I.EQ.L) GO TO 13
      IS=IS+1
      KS=0
      DO 12 K=1, N
      IF(K.EQ.L) GO TO 12
      KS=KS+1
      RNEW(IS, KS)=(R(I, K)-R(I, L)*R(K, L))/
     1  (SQRT(1.-R(I, L)**2)*SQRT(1.-R(K, L)**2))
   12 CONTINUE
   13 CONTINUE
      GRAD(L)=(1./SQRT(6.2831853))*EXP(-X(L)**2/2.)*
     1  PLNORM(RNEW, XNEW, N-1, NMAX, NTRIAL, EPS, IRAND)
    1 CONTINUE
      RETURN
      END
```

4. Számítási eredmények

A programok az angol SCICON *Computer Services Limited Milton Keynes*-i központjának a UNIVAC 1108-as számítógépére lettek elkészítve. Az RNORM függvényeljárást a program könyvtárból vettük, a SUPNOR szubrutin pedig egy külön erre a célre elkészített, gyors, standard normális eloszlású véletlen számokat generáló szubrutin (10 ezer véletlen számot kb. 0,65 másodperc alatt állít elő).

Az eloszlásfüggvény értékek számolási ideje függ az előírt pontosságtól és attól, hogy hány véletlen vektort vizsgálunk egyszerre. Az EPS=0,005 és NTRIAL=250 paraméter értékekkel egy ötdimenziós eloszlásfüggvény értékének a számítási ideje kb. 0,35 másodperc (nagyobb valószínűség számítása esetén általában kevesebb), a gradiens vektor számítási ideje kb. 1,5 másodperc. Tíz dimenziós esetben változatlan pontossággal és NTRIAL=100 paraméter értékekkel az eloszlásfüggvény számítási ideje kb. 0,65 másodperc, a gradiens vektoré kb. 5 másodperc. Egyetlen húsz dimenziós esetet számítottunk, erre a függvényérték számítási ideje 3,3128 másodperc volt, a gradiens vektoré 65,8982 másodperc.

A korreláció mátrix elemeit és az eloszlásfüggvény argumentumait minden alkalommal véletlenszerűen választottuk. (Az argumentumokat 0 és 3 közötti egyenletes eloszlással, a korreláció mátrixokat pedig mint egyenletes eloszlású véletlen vektorok diadikus szorzatainak az összegét). A 2. táblázat öt dimenziós, független normális eloszlású valószínűségi változókra vonatkozó számítások eredményeit tartalmazza. A jelölések azonosak a programban és a blokkdiagramban használtakkal. Ebben az esetben ellenőrizni lehet a számítások tényleges pontosságát is. Érdemes megfigyelni a táblázat utolsó sorát, amely azt mutatja, hogy nem túl nagy valószínűség értékekre is igen pontos lehet a (2.1) átalakítás segítségével számított közelítés (vagy akár az abban foglalt determinisztikus rész), feltéve, hogy az argumentumok között szerepel egyetlen, kiugróan kicsi érték. Azt is észrevehetjük, hogy a két módon nyert közelítés sok esetben közrefogja a valódi értéket, amikor is az interpolált érték általában igen pontos közelítést eredményez.

2. TÁBLÁZAT. Számítási eredmények ötdimenziós, független esetben

	Pontos érték	Végső közelítés	Hiba	PL	S	PH	S+PH	ITER* NTRIAL	Idő
1	0,466 47	0,474 47	0,002 00	0,457 00	0,392 12	0,081 00	0,473 12	1000	0,6758 sec
2	0,867 44	0,866 75	0,000 69	0,812 00	0,862 75	0,004 00	0,866 75	250	0,1718 sec
3	0,561 53	0,559 52	0,002 01	0,548 00	0,505 58	0,064 00	0,569 58	500	0,3408 sec
4	0,909 17	0,907 95	0,001 22	0,912 00	0,907 95	0,000 00	0,907 95	250	0,1696 sec
5	0,808 36	0,806 72	0,001 64	0,804 00	0,798 51	0,009 34	0,807 85	750	0,5062 sec
6	0,902 36	0,899 38	0,002 98	0,896 00	0,899 38	0,000 00	0,899 38	250	0,1692 sec
7	0,726 78	0,729 69	0,002 91	0,737 33	0,704 54	0,021 34	0,725 88	750	0,5018 sec
8	0,487 48	0,488 67	0,001 19	0,508 00	0,383 58	0,086 67	0,470 25	750	0,5050 sec
9	0,377 87	0,384 59	0,006 72	0,394 67	0,179 16	0,193 33	0,372 49	750	0,5014 sec
10	0,533 34	0,533 34	0,000 00	0,538 00	0,456 41	0,073 00	0,529 41	1000	0,6706 sec
11	0,735 83	0,727 89	0,007 94	0,718 00	0,725 88	0,008 00	0,733 88	500	0,3352 sec
12	0,773 75	0,762 73	0,011 02	0,836 00	0,758 73	0,004 00	0,762 73	250	0,1656 sec
13	0,803 86	0,810 65	0,006 79	0,828 00	0,794 59	0,010 00	0,804 59	500	0,3384 sec
14	0,438 53	0,434 49	0,004 04	0,436 00	0,367 57	0,065 00	0,432 57	1000	0,6784 sec
15	0,792 47	0,795 34	0,002 87	0,796 00	0,779 09	0,016 00	0,795 09	750	0,5044 sec
16	0,359 71	0,352 88	0,006 83	0,356 00	0,176 30	0,172 00	0,348 30	750	0,5072 sec
17	0,381 22	0,385 09	0,003 87	0,380 00	0,194 58	0,197 00	0,391 58	1000	0,6786 sec
18	0,114 58	0,112 86	0,001 72	0,110 67	0,646 22	0,766 66	0,120 44	750	0,5098 sec
19	0,389 50	0,391 51	0,002 01	0,420 00	0,177 69	0,182 76	0,360 36	750	0,5076 sec
20	0,574 00	0,570 62	0,003 38	0,520 00	0,570 62	0,000 00	0,570 62	250	0,1766 sec

IRODALOM

[1] DEÁK, I., „A többdimenziós tér halmazai valószínűségeinek kiszámítása normális eloszlás esetén", *Alk. Mat. Lapok* **2** (1976).
[2] PRÉKOPA, A., *Valószínűségelmélet* (Műszaki Könyvkiadó, Budapest, 1962).
[3] PRÉKOPA, A., GANCZER, S., DEÁK, I. és PATYI, K., „A STABIL sztohasztikus programozási modell és annak kísérleti alkalmazása a magyar villamosenergiaiparra", *Alk. Mat. Lapok* **1** (1976) 3—22.

(Beérkezett: 1976. január 15.)

SZÁNTAI TAMÁS
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1502 BUDAPEST XI., KENDE U. 13—17.

AN ALGORITHM FOR CALCULATING MULTIPLE NORMAL DISTRIBUTION FUNCTION VALUES AND GRADIENT VECTOR OF THAT

T. SZÁNTAI

We present an algorithm for calculating the distribution function values of the multivariate normal probability distribution and show how to use it for calculating the gradient vector of that. The algorithm is based on the formula (2.1). We give a flow chart of the algorithm and the list of the program in FORTRAN. The algorithm is very efficient to calculate probability values near to one.

Alkalmazott Matematikai Lapok 2 (1976) 41—55

ADATSZERKEZETEK ABSZTRAKT SZINTAXISA ÉS SZEMANTIKÁJA[1]

VARGA LÁSZLÓ
Budapest

Az adatszerkezeteknek a konkrét ábrázolásuktól független definiálásával foglalkozunk. Definiáljuk az adatelem és az adatszerkezet absztrakt fogalmát. Javaslatot teszünk a *Bécsi Definiciós Nyelv* (VDL) egy olyan kiterjesztésére, amellyel szemléletes módon irható le az adatszerkezetek absztrakt szerkezete és jelentése. Bevezetjük a VDL gráf fogalmát, és annak tulajdonságait tételekbe foglaljuk össze. A definíciós módszert példával szemléltetjük.

1. Bevezetés

A számítástechnikában alkalmazott módszereket numerikus és nem numerikus módszerekre szokás felosztani. Mint ismeretes, a számítástechnika alkalmazásainak túlnyomó többségében nem numerikus módszereket használunk a feladat megoldására. A nem numerikus módszerekben rendszerint bonyolult adatszerkezetekkel végzünk műveleteket. Az adatszerkezetek tulajdonságainak a vizsgálata tehát a számítástechnika alkalmazásai szempontjából nagy jelentőségű.

Ma a különböző programozási rendszerekben gyakran különböző adatszerkezeteket valósítanak meg. Ez megnehezíti a program egy másik rendszerbe való beépítését, megnehezíti a különböző rendszerek programjai között az információcserét. Sőt fennáll az is, hogy nem pontosan azonos adatszerkezeteket azonos névvel jelölünk. Szükség van tehát olyan absztrakt adatszerkezetek definiálására, amelyekből egyértelműen származtathatók a konkrét adatszerkezetek.

A programozási nyelvek gépi reprezentációtól független definiálására az utóbbi évtizedben nagyon nagy energiát fordítottak. Úgy gondolom, ugyanilyen fontos az adatszerkezetek absztrakcióinak megfelelő formális eszközökkel történő szabatos definiálása.

Az adatszerkezetekkel kapcsolatban három probléma merül fel:

— Az adatszerkezetek definiálása.
— Műveletek értelmezése az adatszerkezeteken.
— Az adatszerkezetek és a rajtuk értelmezett műveletek megvalósítása egy konkrét rendszerben.

Ezzel az utóbbi problémával, tehát az adatszerkezetek konkrét reprezentációjának problémájával, most nem foglalkozunk.

Az adatszerkezetekkel kapcsolatos műveletek mind visszavezethetők olyan alapvető elemi műveletekre, amelyek az adatszerkezet lényegéhez tartoznak. Ezeket

[1] Elhangzott az MTA III. Osztály Számítástudományi Bizottsága által rendezett tudományos ülésen, 1975. szeptember 9.-én.

a műveleteket az adatszerkezet definiálásakor meg kell adnunk. Ezeknek a műveleteknek az értelmezése az adatszerkezet definiálásának szerves része. Ebben a munkában az adatszerkezetek definiálásának legfontosabb kérdéseivel foglalkozunk.

Definiáljuk az adatelem és adatszerkezet absztrakt fogalmát. Rámutatunk arra, hogy a VDL *(Bécsi Definíciós Nyelv)* milyen módosításokkal tehető a szemléletesség megtartása mellett alkalmassá különböző típusú adatszerkezetek absztrakcióinak leírására. A szükséges módosítás részleteit csak érintjük.

Az adatszerkezetek szemantikájának egy szemléletes leírását J. EARLY [1] munkájában találhatjuk meg. D. J. ELLIS [5] munkája — azonkívül, hogy a problémakör gazdag irodalmát adja — a hálóelmélet és a lambda kalkulus eszközeivel kísérli meg az adatszerkezetek szemantikájának szabatos leírását. Ezek a munkák jelentősen hozzájárultak az adatszerkezet lényegének a tisztázásához.

C. R. A. HOARE [2] munkájában a gyakorlatban használatos adatszerkezetek egy jó osztályozását találhatjuk meg.

Néhány konkrét adatszerkezet absztrakciójának VDL segítségével történő leírását A. N. LEE [3] munkájában adja meg. A VDL-t tárgyaló más munkákban is találhatunk adatszerkezet leírásokat. Egy ilyen forrásmunka P. WEGNER [4] munkája, amely a VDL kitűnő leírását nyújtja. A *Bécsi Definíciós Nyelv*vel most ismerkedő olvasó számára a [6] dolgozat tanulmányozását ajánljuk.

2. Az adatszerkezet absztrakt fogalma

Mindenekelőtt tisztázzuk az adatelem fogalmát. A programozásban ma a feladat megoldását szintekre tagoljuk. Az egyes szinteken bizonyos összefüggő adathalmazokat elemi adatoknak tekintünk, azokon műveleteket értelmezünk, és ezekkel a műveletekkel írjuk fel a feladat megoldására alkalmas programot. Az adatelem tehát relatív fogalom, amelyről csak egy meghatározott programozási szinten van értelme beszélni.

2.1. DEFINÍCIÓ. Az adatelem egy olyan adategység, amelynek részeit a programozás egy adott szintjén nem definiáljuk, amellyel a programozás adott szintjén műveletek végezhetők, de annak részeivel nem.

Azok az adatok, amelyekkel dolgozunk, rendszerint valamilyen összefüggő halmazát képezik az adatelemeknek. Itt azonban nem az adatelemek közötti tartalmi, hanem hozzáférési, szerkezeti összefüggésekről van szó.

2.2. DEFINÍCIÓ. A szerkezeti összefüggések a halmaz részeihez való hozzáférés módját, sorrendiségét határozzák meg.

Az adatszerkezetekkel különböző műveleteket végzünk. Ezek a műveletek azonban mind visszavezethetők olyan alapvető műveletekre, amelyek a következőképpen csoportosíthatók:

— szelekciós műveletek,

— konstrukciós műveletek.

2.3. DEFINÍCIÓ. A szelekciós műveletek olyan leképezések, amelyek az adatszerkezetet annak részeire képezik le.

2.4. DEFINÍCIÓ. A konstrukciós műveletek olyan tevékenységek, amelyekkel az adatszerkezet felépíthető, illetve módosítható.

A szelekciós és konstrukciós műveletek az adatszerkezet lényegéhez tartoznak, mert ezekkel a műveletekkel lehet az adatszerkezet jelentését visszavezetni az azt alkotó adatelemek jelentésére.

2.5. DEFINÍCIÓ. Az adatszerkezet az adatelemeknek egy olyan összefüggő halmaza, amelyhez hozzá vannak rendelve

— a szerkezeti összefüggések,
— a szelekciós műveletek,
— a konstrukciós műveletek.

Formálisan az adatszerkezet egy (A, R, S, K) négyes, ahol

A az adatelemek
R a szerkezeti összefüggések
S a szelekciós műveletek
K a konstrukciós műveletek

véges halmaza.

Az adatszerkezetek definiálásához olyan formális eszközre van szükségünk, amellyel ez a négyes leírható. Ilyen formális eszköznek látszik a VDL. VDL objektumokkal definiálhatjuk a szerkezeti összefüggéseket, és az absztrakt gépre megadhatjuk a szelekciós és a konstrukciós műveleteket. Felmerülnek azonban a következő kérdések:

1. Meg lehet-e választani olyan alapobjektumokat, amelyekből az adatszerkezeteknél fellépő bármely szerkezeti összefüggés leírására alkalmas objektum felépíthető?

2. Hogyan célszerű megválasztani ezeket az alapobjektumokat?

3. Az alapobjektumokon milyen szelekciós és konstrukciós műveleteket kell értelmezni ahhoz, hogy azokból bármely adatszerkezet szelekciós és konstrukciós algoritmusai felépíthetők legyenek?

Az első kérdésre könnyű válaszolni. Az adatszerkezeteket a számítógépek memóriájában, háttértáraiban ábrázoljuk. Ha tehát ezeknek az ábrázolásoknak az absztrakcióit alapobjektumoknak választjuk meg, akkor ezekből az objektumokból felépíthetők a gyakorlat számára fontos adatszerkezetek leírására alkalmas objektumok. Ez a módszer azonban bonyolult, nehézkes módszer, és főleg nem szemléletes.

Az alapobjektumok célszerű megválasztása sokkal nehezebb feladat. Ha csak elemi alapobjektumokat választunk meg, például a VDL elemi objektumait, akkor az adatszerkezeteket leíró objektumok elveszítik szemléletes jelentőségüket, áttekintésük nehézkessé válik. Ha viszont túl sok összetett alapobjektumot vezetünk be, akkor azok használata megnehezíti a különböző adatszerkezetek egységes leírását, akadályozza azok összehasonlító elemzését.

A harmadik kérdés megválaszolásakor hasonló problémákkal találkozunk.

Ebben a tanulmányban ezekre a kérdésekre próbáljuk megadni a választ gyakorlati megfontolás alapján. Természetesen ennek a tanulmánynak a keretében nem térhetünk ki a fenti kérdésekre adandó válasz minden részletére.

3. Alapobjektumok

A gyakorlatban az adatelemek halmazának általában háromféle ábrázolásával találkozunk. Ezek a következők:

— indexelt ábrázolás,
— szekvenciális ábrázolás,
— láncolt ábrázolás.

Kézenfekvő gondolat ezeket az ábrázolási formákat az adatszerkezetek alapvető szerkezeti összefüggéseinek megvalósításaiként felfogni. Ekkor az alapobjektumok a fenti ábrázolási formák absztrakciói lehetnek. Így az alapobjektumok három osztályához jutunk, amelyek közül az első kettő a VDL-ben megtalálható. Ezeken az alapobjektumokon azokat a műveleteket értelmezhetjük, amelyeket a VDL-ben bevezettek.

3.1. Axióma. Indexelt ábrázolás esetén az adatelemek egy olyan halmazt alkotnak, amelyek minden eleméhez kölcsönösen és egyértelműen hozzá van rendelve egy azonosító, amellyel az a halmazból kiválasztható.

3.1. Definíció. Az indexelten ábrázolt adatok halmazainak absztrakciói olyan objektumok, amelyeknek halmazát a következő predikátum határozza meg:

$$\text{is}-\text{data}-\text{set} = (\{\langle s{:}\text{is}-\text{data}\rangle | \text{is}-\text{selector}(s)\})$$

Az is−selector predikátum által meghatározott halmaz nem lényeges itt. Lehet ez tetszőleges szimbólumok, nevek stb. halmaza. Az is−data predikátum olyan objektumok halmazát határozza meg, amelyeket az adott definíciós szinten elemi objektumoknak tekintünk.

Állapodjunk meg a következő jelölésben:

$$n \in g$$

akkor, és csak akkor, ha

$$\text{is}-\text{data}-\text{set}(g) = \text{TRUE} \quad \text{és} \quad (\exists s)\,(s(g) = n).$$

Az is−data−set predikátumnak eleget tevő objektumot az 1. ábrán látható rajzzal szemléltethetjük.

Ha $\text{is}-\text{data}-\text{set}(t)=\text{TRUE}$, akkor azon értelmezve vannak a következő műveletek:

$$s(t),$$

$$\mu(t; \langle s{:}a\rangle),$$

ahol

$$\text{is}-\text{selector}(s) \wedge (\text{is}-\text{data}(a) \vee \text{is}-\text{'NIL'}(a)) = \text{TRUE}.$$

Ezeket a műveleteket itt nem definiáljuk, csupán az irodalomra utalunk [3].

3.2 Axióma. A szekvenciálisan ábrázolt adatok sorba rendezettek, ezért a halmaz elemei sorszámukkal azonosíthatók.

3.2. DEFINÍCIÓ. A szekvenciálisan ábrázolt adatok halmazainak absztrakciói olyan objektumok, amelyeknek halmazát a következő predikátum határozza meg:

$$\text{is}-\text{data}-\text{list} = (\{\langle \text{elem}(i):\text{is}-\text{data}\rangle | 1 \leqq i \leqq n\}),$$

ahol n a halmaz elemeinek száma. A VDL listát a 2. ábra mutatja.

Ha is−data−list(t)=TRUE, akkor azon értelmezve vannak a következő műveletek:

length (t),
elem (i) (t),
head (t),
tail (t),
$\mu(t; \langle \text{elem}(i):a\rangle)$,
delete (t, i),

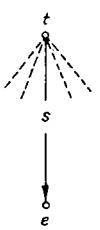
ahol is−data(a)=TRUE, $1 \leqq i \leqq n$.

Ha is−data−list(t_1)∧is−data−list(t_2)=TRUE, akkor azokon értelmezzük a

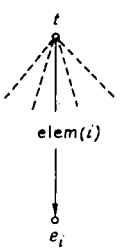
$$t_1 \frown t_2$$

műveletet is.

A műveletek definiálásával szintén nem kívánunk foglalkozni. A műveletek értelmezése a [3] munkában megtalálható.



1. ábra



2. ábra

3.3. AXIÓMA. Láncoltan ábrázolt adatelemek olyan halmazt alkotnak, amelyben az adatelemekhez hozzá vannak rendelve azoknak az adatelemeknek az azonosítói is, amelyek az adott adatelemtől közvetlenül elérhetők. A halmaznak van legalább egy olyan kitüntetett adateleme, amely közvetlenül elérhető, míg a halmaz további adatelemei csak a kitüntetett adatelemektől kiindulva érhetők el. A halmazt alkotó adatelemek ily módon bejárhatók.

3.3. DEFINÍCIÓ. A láncoltan ábrázolt adatok halmazainak absztrakciói olyan objektumok, amelyeknek halmazát a következő predikátum határozza meg:

is−data−graph=is−node−set
is−node=($\langle s-\text{value}:\text{is}-\text{data}\rangle$, $\langle s-\text{desc}:\text{is}-\text{selector}-\text{list}\rangle$),

ahol ha is−data−graph(g)=TRUE, akkor ahhoz hozzá van rendelve egy nem üres

$M = \{n|\text{is}-\text{master}(n)\} \subset \{n|\text{is}-\text{node}(n)\}$

halmaz úgy, hogy valahányszor

$t \in g$ és is−master(t)=FALSE,

mindannyiszor t *elérhető* (lásd 3.5. definíció), legalább egy olyan m-ből, amelyre is−master(m) = TRUE.

3.4. DEFINÍCIÓ. Legyen is−data−graph(g)=TRUE, $t \in g$, $n \in g$.
Akkor és csak akkor mondjuk azt, hogy az n *hivatkozik* a t-re, ha

$$(\exists i,\ 1 \leqq i \leqq \text{length}(s-\text{desc}(n)))\,(\text{elem}(i)\,(s-\text{desc}(n))\,(g) = t).$$

Jelölése:

$$n \rightarrow t$$

3.5. DEFINÍCIÓ. Legyen is−data−graph(g)=TRUE.
Akkor és csak akkor mondjuk, hogy t_1-ből *elérhető* t_k, ha

$$t_1 \rightarrow t_2 \rightarrow \ldots \rightarrow t_k,$$

ahol $t_i \in g$, $i=1, 2, \ldots, k$.

Ha t_1-ből elérhető t_k, akkor más szavakkal azt modjuk, hogy t_1-ből *hivatkozási út vezet* t_k-ba. Jelölése:

$$t_1 \rightarrow * t_k$$

Megjegyzés: A 3.3. definíció alapján a következőt állíthatjuk:
Legyen is−data−graph(g)=TRUE.
Ha is−master(t)=FALSE és $t \in g$, akkor

$$(\exists m \in g)(m \rightarrow * t \text{ és } \text{is-master}(m)).$$

3.6. DEFINÍCIÓ. Legyen is−data−graph(g)=TRUE.
Akkor és csak akkor mondjuk, hogy $n \in g$ terminális, ha

$$s-\text{desc}(n) = \langle\rangle.$$

3.7. DEFINÍCIÓ. Legyen

$$\text{value}(n) = \begin{cases} s\text{-value}(n), & \text{ha} \quad \text{is-node}(n) = \text{TRUE}, \\ \text{NIL}, & \text{ha} \quad \text{is-node}(n) = \text{FALSE} \end{cases}$$

és

$$\text{next}(i)(n) = \begin{cases} (\text{elem}(i)(s\text{-desc}(n)))(g), & \text{ha} \quad \text{is-data-graph}(g) = \text{TRUE} \quad \text{és} \quad n \in g, \\ \text{NIL}, & \text{különben.} \end{cases}$$

A gráf objektumot a 3. ábra mutatja. Ez az ábrázolási forma azonban nem fejezi ki szemléletesen a gráf szerkezetét. A fent bevezetett függvények segítségével azonban ábrázolhatjuk a gráfot szemléletes módon is. Ezt mutatja az 5. ábra a 4. ábrán látható konkrét gráf esetében, ahol a gráf közvetlenül elérhető (master) elemei az s_1 és s_4 szelektorokkal választhatók ki.

A gráffal kapcsolatban a legfontosabb művelet a gráf bejárása. Konstruáljuk meg a gráf bejárását elvégző absztrakt gépet. Határozza meg a gép ξ_i ($i=0, 1, 2, \ldots$) állapotainak halmazát az

$$\begin{aligned} \text{is}-\text{state} = (&\langle s-\text{input}:\text{is}-\text{data}-\text{graph}\rangle, \\ &\langle s-\text{table}:\text{is}-\text{table}\rangle, \\ &\langle s-\text{control}:\text{is}-\text{control}\rangle) \end{aligned}$$

predikátum, ahol

$$\text{is}-\text{table} = (\{\langle s:\text{is}-\text{value}\rangle | \text{is}-\text{selector}(s)\})$$

$$\text{is}-\text{value} = \{T, F\}.$$

Legyen

$$\xi_0 = \mu_0(\langle s-\text{input}:g\rangle, \langle s-\text{table}:t_0\rangle, \langle s-\text{control}:\underline{\text{walk}}(g)\rangle),$$

ahol

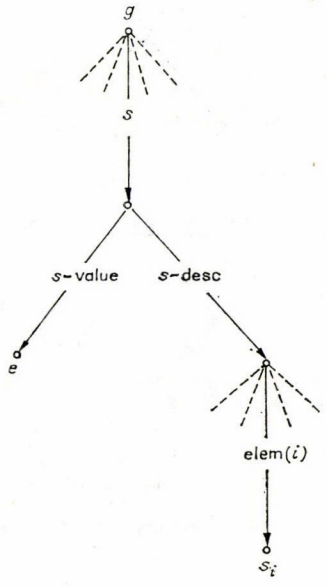
$$\text{is}-\text{data}-\text{graph}(g) = \text{TRUE}$$

és

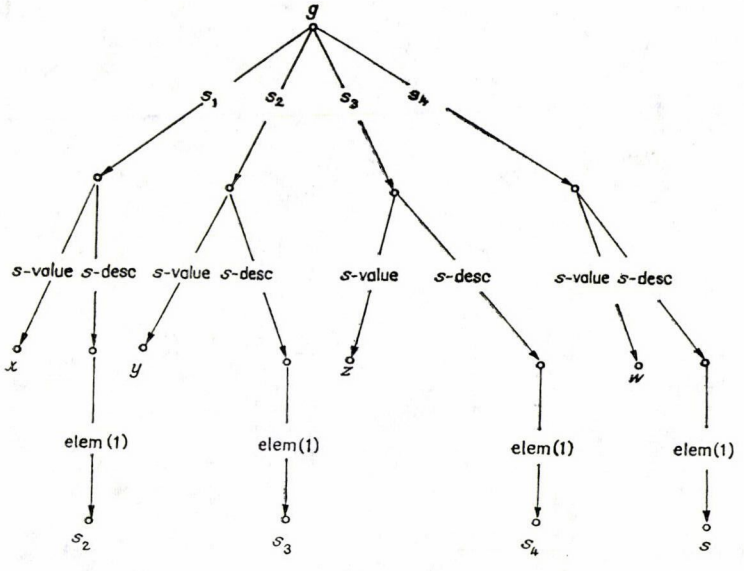
$$t_0 = (\{\langle s:F\rangle | \text{is-master}(s(g))\}).$$

3.8. DEFINÍCIÓ. Legyen $\text{next}-\text{selector}(t)$ az a függvény, amelynek értelmezési tartománya a

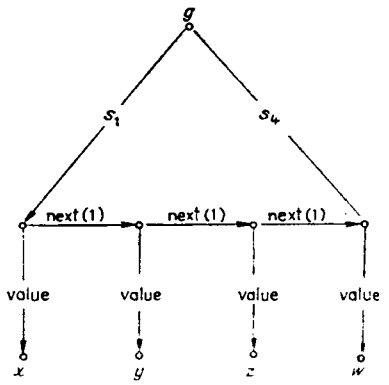
$$\{t \mid \text{is-table}(t)\}$$



3. ábra



4. ábra



5. ábra

halmaz és értékkészlete pedig

$$\{s \mid \text{is-selector}(s)\} \cup \{\text{NIL}\}.$$

Ha $\text{is}-\text{table}(t)=\text{TRUE}$ és

$$(\exists s)(s(t) = F)$$

akkor legyen $\text{next}-\text{selector}(t)$ egy olyan szelektor, amelyre

$$\text{next-selector}(t)F = (t)$$

különben legyen

$$\text{next-selector}(t) = \text{NIL}.$$

3.1. LEMMA. Legyen $\text{is}-\text{data}-\text{list}(t)=\text{TRUE}$. Akkor a

$$\underline{\text{process}-\text{list}}(t) =$$

$$\text{length}(t) = 0 \rightarrow \underline{\text{null}}$$

$$T \rightarrow \underline{\text{process}-\text{list}}(\text{tail}(t));$$

$$\underline{\text{process}}(\text{head}(t))$$

művelet a t lista minden elemére egyszer és csak egyszer alkalmazza a *process* operációt.

Bizonyítás. Az állítás helyessége közvetlenül adódik a head és tail műveletek értelmezéséből.

3.1. TÉTEL. Legyen ξ_0 a fent megadott. Akkor a következő algoritmus minden

$$n \in g$$

objektumra egyszer és csak egyszer alkalmazza a $\underline{\text{process}}(\text{value}(n))$ operációt.

$$\underline{\text{walk}-\text{graph}}(g) =$$

$$\text{next}-\text{selector}(s-\text{table}(\xi)) = \text{NIL} \rightarrow \underline{\text{null}}$$

$$T \rightarrow \underline{\text{walk}-\text{graph}}(g);$$

$$\underline{\text{process}}(v);$$

$$v: \underline{\text{next}-\text{value}}(n);$$

$$n: \underline{\text{next}-\text{node}}(s);$$

$$s: \underline{\text{pass}}(\text{next}-\text{selector}(s-\text{table}(\xi)))$$

pass(t) =

PASS:t

next−node(s) =

PASS:$s(s-\text{input}(\xi))$

$s-\text{table}:\mu(s-\text{table}(\xi);\langle s:T\rangle)$

next−value(n) =

pass(value(n)),

process−selector($s-\text{desc}(n)$)

process−selector(list) =

length(list) = 0→null

T→process−selector(tail(list));

set(head(t))

set(s) =

$s(s-\text{table}(\xi))$ = NIL→link(s)

T→null

link(s) =

$s-\text{table}:\mu(s-\text{table}(\xi);\langle s:F\rangle)$.

Bizonyítás. Bizonyítsuk be, hogy a walk−graph(g) vezérlési fája akkor és csak akkor redukálódik a null utasításra, ha minden $n\in g$ mellett a

$$v = \text{value}(n)$$

objektumra pontosan egyszer alkalmazásra került a process(v) operáció.

Tegyük fel először azt, hogy a vezérlési fa a null utasításra redukálódott. Ekkor a 3.8. definíció értelmében nincs olyan s, amelyre

$$s(s\text{-table}(\xi)) = F.$$

Ez csak úgy lehet, ha minden olyan s mellett, amelyre

$$\text{is}-\text{master}(s(g)) = \text{TRUE}$$

végrehajtásra került a next−node(s) utasítás. Ennek eredményeként az adott s mellett

$$s(s\text{-table}(\xi)) = T$$

áll elő, másrészt végrehajtódik a

next−value($s(g)$)

utasítás. Ennek az utasításnak az a következménye, hogy egyrészt végrehajtódik a

$$\underline{\text{process}}(\text{value}(s(g)))$$

utasítás, másrészt minden olyan s szelektor, amellyel $s(g)$ hivatkozik egy $n \in g$-re

a) ha szerepel a táblázatban, értéke változatlan marad,

b) ha nem szerepel a táblázatban, akkor az oda F értékkel kerül fel.

A b) pont alapján nyilvánvaló, hogy ha a vezérlési fa a null utasításra redukálódott, akkor szükségképpen feldolgozásra került minden olyan $n \in g$, amelyre

$$m \rightarrow * n, \quad \text{is}-\text{master}(m) = \text{TRUE}.$$

Ámde a 3.3. definíció értelmében ez azt is jelenti, hogy minden $n \in g$ feldolgozásra került. Az a) pont alapján és a vezérlési fa alapján az is nyilvánvaló, hogy minden $n \in g$ mellett pontosan egyszer hajtódik végre a

$$\underline{\text{process}}(\text{value}(n))$$

utasítás.

Most tegyük fel, hogy minden $s(g) \neq \text{NIL}$ mellett végrehajtásra került a

$$\underline{\text{process}}(\text{value}(s(g)))$$

utasítás, és mutassuk meg, hogy akkor a vezérlési fa a null utasításra redukálódik.

Minden $\underline{\text{process}}(\text{value}(s(g)))$ utasítást meg kell előznie a $\underline{\text{next}-\text{node}}(s(g))$ utasításnak, aminek eredményeként

$$s(s\text{-table}(\xi)) = T$$

áll elő, és ez az érték a táblázatban változatlan marad. Mivel a set utasítás csak olyan s szelektort visz fel a táblázatba, amelyre $s(g) \neq \text{NIL}$, ezért minden $s(g) \neq \text{NIL}$ objektum feldolgozása után a táblázatban minden szelektor értéke T lesz, és a vezérlési fa a null utasításra redukálódik. Ezzel a 3.1. tétel bizonyítása teljes.

A bevezetett szelekciós műveletek segítségével felírható tehát a gráffal kapcsolatos egyik legfontosabb algoritmus, a gráf bejárásának algoritmusa. Természetesen a gráf bejárására többféle stratégia létezik. A 3.1. tétel egy olyan bejárási algoritmust ad meg, amelyből különböző bejárási stratégiák algoritmusai származtathatók a next−selector függvény konkrét megválasztásával.

A gráf bejárási algoritmusának egyik alkalmazása a szegmensekből álló program összeszerkesztése. Ennek részletes kidolgozása a szerző [7] munkájában található meg.

A gráffal kapcsolatos műveletek egy része konstrukciós művelet. Ezeknek a felírásához konstrukciós alapműveletekre van szükségünk. Ilyen konstrukciós alapműveletek a gráf bejárási stratégiájára épülhetnek fel. Ezeknek az alapműveleteknek a definiálásától az általános esetben most eltekintünk, csupán a gráf egy fontos speciális esetében adjuk meg azokat. Ez a speciális eset a lineárisan láncolt adatszerkezet, a *lánc*.

3.4. Axióma. A lineárisan láncolt adatszerkezet egy olyan adatgráf, amelynek pontosan egy közvetlenül elérhető master és egy terminális eleme van, továbbá minden elem csak egy elemre hivatkozik.

3.9. DEFINÍCIÓ. A lineárisan láncolt adatszerkezetek absztrakciói olyan objektumok, amelyeknek halmazát a következő predikátum határozza meg:

$$\text{is}-\text{data}-\text{chain} = \text{is}-\text{data}-\text{graph}$$

ahol, ha $\text{is}-\text{data}-\text{chain}(g)=\text{TRUE}$, akkor
1. egy és csakis egy olyan $m \in g$ van, amelyre

$$\text{is}-\text{master}(m) = \text{TRUE},$$

2. egy és csakis egy olyan $v \in g$ van, amelyre

$$(\text{elem}(1)\,(s-\text{desc}(v)))\,(g) = \text{NIL},$$

3. ha $n \in g$ és $\text{is}-\text{master}(n)=\text{FALSE}$, akkor

$$(\exists m)\,(m \twoheadrightarrow * t \quad \underline{\text{és}} \quad \text{is}-\text{master}(m)).$$

4. ha $n \in g$ nem terminális, akkor

$$\text{length}\,(s\text{-desc}(n)) = 1.$$

3.2. TÉTEL. Legyen $\text{is}-\text{data}-\text{chain}(g)=\text{TRUE}$,

$$\text{is}-\text{master}(m) = \text{TRUE}, \quad m \in g,$$

$$\text{next}(1)\,(v) = \text{NIL}, \quad v \in g, \quad m \neq v,$$

akkor egy és csakis egy hivatkozási út vezet m-ből v-be.

Bizonyítás. A 3.9. definíció alapján

$$m \twoheadrightarrow * v$$

fennáll. Azt kell tehát csak bizonyítani, hogy ez egyértelműen meghatározott. Ez azonban nyilvánvalóan következik abból, hogy

$$\text{length}(s\text{-desc}(n)) = 1$$

minden olyan $n \in g$-re, amely nem terminális.

3.10. DEFINÍCIÓ. Legyen $\text{is}-\text{data}-\text{chain}(g)=\text{TRUE}$. Értelmezzük a következő műveleteket:
1. Legyen first az a szelektor, amelyre

$$\text{is}-\text{master}(\text{first}(g)\,(g)) = \text{TRUE}.$$

2. Ha $n \in g$, akkor legyen

$$\text{next}(n) = \text{next}(1)\,(n).$$

3. Ha $s(g) \in g$ és $n=\text{next}(s(g))$, akkor

$$\text{cancel}(n, g) =$$

$$\mu(g;\langle s-\text{desc}.s : s-\text{desc}(n)\rangle, \langle \text{elem}(1)\,\langle s-\text{desc}(s(g))\rangle : \text{NIL}\rangle).$$

4. Ha $\text{is}-\text{data}(x)=\text{TRUE}$ és $s(g)\in g$, akkor

$$\text{put}-\text{next}(x, s(g), g) =$$

$$\mu(g; \langle s' : \mu_0(\langle s-\text{value}:x\rangle, \langle s-\text{desc}:s-\text{desc}(s(g))\rangle)\rangle,$$

$$\langle s-\text{desc}.s:\langle s'\rangle\rangle),$$

ahol $s'(g)=\text{NIL}$.
A 3.10. definíció alapján nyilvánvaló, hogy

$$\text{is}-\text{data}-\text{chain}\,(\text{cancel}(n, g)) = \text{TRUE},$$

$$\text{is}-\text{data}-\text{cahin}(\text{put}-\text{next}(x, s(g), g)) = \text{TRUE}.$$

A bevezetett alapobjektumok segítségével most már az adatszerkezet absztrakt fogalmát részletesebben is meg tudjuk fogalmazni.

3.11. DEFINÍCIÓ. Jelölje az adatelemek halmazát

$$\text{is}-\text{datâèlement}.$$

Az adatszerkezet absztrakciója egy olyan (t, S, K) hármassal írható le, ahol

$$\text{is}-\text{data}-\text{structure}(t) = \text{TRUE}$$

$$\text{is}-\text{data}-\text{structure} =$$

$$\text{is}-\text{data}-\text{set}\vee\text{is}-\text{data}-\text{list}\vee\text{is}-\text{data}-\text{graph}$$

$$\text{is}-\text{data} = \text{is}-\text{data}-\text{structure}\vee\text{is}-\text{dataelement}$$

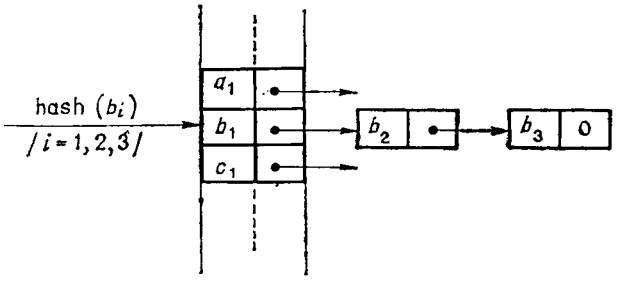
S a t objektumon értelmezett szelekciós műveletek vezérlési fáinak halmaza, K a t objektumon értelmezett konstrukciós műveletek vezérlési fáinak halmaza.

Az $\text{is}-\text{data}-\text{structure}$ predikátumnak eleget tevő objektum az adatszerkezet absztrakt szintaxisát adja meg. Azokat a szerkezeti összefüggéseket definiálja, amelyek az adatszerkezetek ábrázolása és a rajta megvalósítandó műveletek szempontjából lényegesek.

A szelekciós, illetve a konstrukciós műveletek azokat a tevékenységeket definiálják, amelyek az adatszerkezet lényegét, szemantikáját határozzák meg.

4. Példa egy konkrét adatszerkezet formális leírására

Példaként definiáljuk azoknak az indextáblázatoknak a halmazát, amelyeknek altáblázatai lineárisan láncolt adatszerkezetet alkotnak, és az altáblázat indexét egy hash függvény segítségével határozhatjuk meg. Egy ilyen táblázatot szemléltet a 6. ábra.



6. ábra

4.1. DEFINÍCIÓ. Legyen adott az

$$\text{is}-\widehat{\text{argumentum}}$$

és az

$$\text{is}-\widehat{\text{val}}$$

halmaz. A hash-táblázatok absztrakt szintaxisainak halmazát a következő predikátum határozza meg:

$$\text{is}-\text{hash}-\text{table} = \text{is}-\text{subtable}-\text{set}$$

$$\text{is}-\text{subtable} = \text{is}-\text{elem}-\text{chain}$$

$$\text{is}-\text{elem} = (\langle s-\text{arg}:\text{is}-\text{argumentum}\rangle,$$

$$\langle s-\text{val}:\text{is}-\text{val}\rangle).$$

4.2. DEFINÍCIÓ. Legyen $\text{is}-\text{hash}-\text{table}(t)=\text{TRUE}$,

$$a \in \text{is}-\widehat{\text{argumentum}}$$

és legyen adott a $\text{hash}(a)$ függvény, amelyre

$$\text{is}-\text{selector}(\text{hash}(a)) = \text{TRUE}.$$

Ekkor a t táblázat szemantikáját a következő szelekciós és konstrukciós műveletekkel definiálhatjuk:

4.3. DEFINÍCIÓ. Egy adott x argumentumú elem értékének a kiolvasása a t táblázatból:

$$\underline{\text{read}}(x, t) = \underline{\text{test}}(x, \text{first}(\text{hash}(x)(t)))$$

$$\underline{\text{test}}(x, n) =$$

$$n = \text{NIL} \rightarrow \underline{\text{out}}(\text{NIL})$$

$$s-\text{arg}(\text{value}(n)) = x \rightarrow \underline{\text{out}}(s-\text{val}(\text{value}(n)))$$

$$T \rightarrow \underline{\text{test}}(x, \text{next}(n))$$

$$\underline{\text{out}}(t) =$$

$$s-\text{output}:t$$

feltéve, hogy a kiolvasott eredményt az absztrakt gép $s-\text{output}$ (ξ) komponense tartalmazza.

4.4. DEFINÍCIÓ. Egy adott

$$x = \mu_0(\langle s-\text{arg}:a\rangle, \langle s-\text{val}:v\rangle)$$

objektum elhelyezése a t táblázatban:

$$\underline{\text{write}}(x, t) =$$

$$\underline{\text{search}-\text{end}}(x, \text{first}(\text{hash}(x)(t)), t);$$

$$\underline{\text{search}-\text{end}}(x, n, t) =$$

$$\text{next}(n) = \text{NIL} \rightarrow \text{put}-\text{next}(x, n, t)$$

$$T \rightarrow \underline{\text{search}-\text{end}}(x, \text{next}(n), t).$$

4.5. DEFINÍCIÓ. A hash-tábla absztrakciója (T, S, K), ahol

$$\text{is}-\text{hash}-\text{table}(t) = \text{TRUE},$$

$$S = \{\underline{\text{read}}(x, t)\}$$

$$K = \{\underline{\text{write}}(x, t).$$

Hasonlóképpen felírhatók más adatszerkezetek absztrakciói is a bevezetett formális eszközök segítségével. Természetesen nincs objektív kritériumunk az alapobjektumok megválasztására. Mi az említett gyakorlati érvek alapján szükségesnek és hasznosnak véljük a gráfnak, mint összetett objektumnak a bevezetését a VDL-be.

Végül köszönetemet fejezem ki DÖMÖLKI BÁLINTnak és FIDRICH ILONÁnak, akik a kézirat több hiányosságára felhívták figyelmemet.

IRODALOM

[1] EARLY, J., "Toward an understanding of data structures", *Communications of ACM* **14** (1971) 617—627.
[2] HOARE, C. R. A., *Notes on data structuring, Structured programming* (Academic Press, 1972) 83—174.
[3] LEE, A. N., *Computer semantics, Computer Science Series*, (Van Nostrand Reinhold Company, 1972)
[4] WEGNER, P., "The Vienna Definition Language", *Computing Surveys* **4** (1972) 5—63. (Magyar nyelven megjelent a NIM IGÜSZI kiskönyvtár „*Számológép*" című folyóirat 1973. évi számában LANGER TAMÁS fordításában.)
[5] ELLIS, D. J., "Semantics of data structures and references", MC 16 375 (1974) 1—172.
[6] NEUHOLD, E. J., "The formal description of programming languages", *IBM System Journal* **10** (1971).
[7] VARGA, L., "The abstractions of machine dependent program forms", KFKI-76-11 (1976) 1—20.

(Beérkezett: 1976. március 16.)

VARGA LÁSZLÓ
MTA KÖZPONTI FIZIKAI KUTATÓ INTÉZET
1525 BUDAPEST, 114. POSTAFIÓK 49.

ABSTRACT SYNTAX AND SEMANTICS OF DATA STRUCTURE

L. VARGA

The definition problems of data structures, independently of their concrete representations, are discussed.

The abstract concepts of data element and data structures are defined. The *Vienna Definition Language* can be used for defining the abstract syntax and semantics of data structures. The basic VDL data structures are discussed from the viewpont of practical considerations and the VDL graph, as a basic VDL data structure and graph manipulation operators are introduced. The properties of the VDL graph are summarized in theorems. The use of VDL graph in the definition of data structures is illustrated by an example.

Alkalmazott Matematikai Lapok **2** *(1976) 57—66*

AZ *M* MAXIMÁLIS HATÁRÉRTÉK-LOGIKÁRÓL

DEMETROVICS JÁNOS
Budapest

Az Sz. V. JABLONSZKIJ által bevezetett határérték-logika fogalom [2,8] lehetővé teszi, hogy egyidejűleg tárgyaljuk a végesértékű logikák [1,5] bizonyos problémáit. Számos dolgozatban [2,7] vizsgálják a határérték-logikák halmazát néhány természetesen felvetődő parciális rendezés segítségével. Ezek a parciális rendezések különböző morfizmusokon alapultak. Bizonyítást nyert az a tény, hogy ezeknek a parciális rendezéseknek nincs sem minimális, sem maximális elemük.

A [8] dolgozatban bevezetett logikai úton nyert parciális rendezés a határérték-logikák halmazát úgy osztja fel ekvivalens osztályokra, hogy közöttük már létezik maximális határérték-logikák osztálya. Maximális abban az értelemben, hogy minden más határérték-logika képét tartalmazza. Jelen dolgozatban megvizsgáljuk egy *M* reprezentálását ennek a maximális határérték-logikák osztályának.

Dolgozatunkban bebizonyítjuk, hogy az *M* maximális határérték-logika esetén a partíció tartó, a monoton és az önduális típusú zártosztályok mindegyike kontinuum számosságú. Szükséges és elégséges feltételt adunk arra vonatkozóan, hogy az U^R partíció tartó (monoton, önduális) osztály mikor majdnem teljes az *M* határérték-logikában. Megmutatjuk, hogy *M* esetében nem igaz a *Kuznyecov* tétel [1] és konstrutív módon felépítunk egy zártosztályt, amelyet nem lehet kibővíteni majdnem teljes osztállyá.

1. Alapfogalmak

A következőkben definiáljuk azokat az alapfogalmakat, amelyekre a dolgozatban szükségünk lesz.

1.1. DEFINÍCIÓ. Legyen E_k egy tetszőleges k-elemű halmaz ($k \geqq 2$). Jelölje P_k^n ($n=0, 1, 2, \ldots$) az olyan n változós $f(x_1, x_2, \ldots, x_n)$ függvények halmazát, amelyek változói és értékei az E_k halmaz elemei. k-értékű logikának nevezzük a $P_k = \bigcup_{n=0}^{\infty} P_k^n$ függvényhalmazt. Az általánosság megszorítása nélkül feltehetjük, hogy $E_k = \{0, 1, \ldots, k-1\}$.

1.2. DEFINÍCIÓ. Ha az 1.1. definícióban szereplő E_k véges halmazt valamilyen $\aleph_0$ számosságú $E_{\aleph_0}$ halmazzal helyettesítjük, akkor az így kapott $P_{\aleph_0}$ függvényhalmazt végtelenértékű logikának nevezzük. Dolgozatunkban $E_{\aleph_0}$ mindig a nem negatív egész számok halmazát jelöli.

A [2] dolgozatban megtalálható a zárt függvényosztály, a homomorf leképzés, az izomorfizmus és a szuperpozíció definíciója.

1.3. DEFINÍCIÓ. A $P_{\aleph_0}$ végtelen értékű logika P részhalmazát határérték-logikának nevezzük, ha teljesülnek a következő feltételek:

a) P függvényosztályban csak megszámlálható sok függvény van;

b) minden természetes k számhoz ($k \geqq 2$) létezik egy olyan függvényhalmaz ($A_k \subseteq P$), amelyet homomorf módon le lehet képezni a k-értékű logikára.

1.4. DEFINÍCIÓ. Legyen $\mathfrak{M} \subseteq \mathfrak{N} \subseteq P_{\aleph_0}$. Az $\mathfrak{M}$ részhalmaz teljes az $\mathfrak{N}$ függvényhalmazban, ha $[\mathfrak{M}] = \mathfrak{N}$. Az $\mathfrak{M}$ függvényhalmaz majdnem teljes az $\mathfrak{N}$ függvényhalmazban, ha nem teljes az $\mathfrak{N}$ halmazban, de ha hozzá adunk bármilyen $f(x_1, x_2, \ldots, x_n)$ függvényt az $\mathfrak{N} \setminus \mathfrak{M}$ halmazból, akkor az $[\{f(x_1, x_2, \ldots, x_n)\} \cup \mathfrak{M}]$ teljes lesz az $\mathfrak{N}$ függvényhalmazban.

1.5. DEFINÍCIÓ. Egy B függvénysorozatot a Q függvényhalmaz bázisának nevezzük, ha teljesülnek a következő feltételek:

a) B teljes a Q függvényhalmazban, azaz $[B] = Q$;

b) nincs a B halmaznak olyan valódi részhalmaza, amely a Q halmazban teljes.

Legyen $s(x)$ egy olyan permutáció, amely véges sok értéket permutál és $s(0) = 0$ $(s(x) \in P_{\aleph_0})$.

1.6. DEFINÍCIÓ. Az

$$f^{s(x)}(x_1, x_2, \ldots, x_n) = s^{-1}\big(f\big(s(x_1), s(x_2), \ldots, s(x_n)\big)\big)$$

függvény az $f(x_1, x_2, \ldots, x_n)$ függvény duálisa az $s(x)$ függvényhez viszonyítva.

1.1. *Megjegyzés.* Ha az $f(x_1, x_2, \ldots, x_n)$ függvény az $f_1(x_{11}, x_{12}, \ldots, x_{1m_1})$, $f_2(x_{21}, x_{22}, \ldots, x_{2m_2})$, $\ldots, f_n(x_{n_1}, x_{n_2}, \ldots, x_{nm_n})$ függvények szuperpozíciójának az eredménye, akkor az $f^{s(x)}(x_1, x_2, \ldots, x_n)$ függvény az $f_1^{s(x)}(x_{11}, x_{12}, \ldots, x_{1m_1}), f_2^{s(x)}(x_{21}, x_{22}, \ldots, x_{2m_2}), \ldots, f_n^{s(x)}(x_{n1}, x_{n2}, \ldots, x_{nm_n})$ függvények szuperpozíciójának az eredménye ugyanabban a sorrendben.

1.7. DEFINÍCIÓ. Az $f(x_1, x_2, \ldots, x_n)$ függvény önduális az $s(x)$ függvényhez viszonyítva, ha $f^{s(x)}(x_1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, x_n)$.

1.8. DEFINÍCIÓ. Legyen r az $E_{\aleph_0} \setminus 0$ halmaz parciális rendezése $(\prec_r) \cdot \tilde{\alpha} \prec_r \tilde{\beta}$, ha $\forall_i (i = 1, 2, \ldots, n) \alpha_i \prec_r \beta_i$. Azt mondjuk, hogy $f(\tilde{x})$ monoton függvény az $\prec_r$ parciális rendezéshez viszonyítva, ha minden $\tilde{\alpha}$ és $\tilde{\beta}$ szám n-es párra, amelyre igaz, hogy $\tilde{\alpha} \prec_r \tilde{\beta}$, igaz az is, hogy $f(\tilde{\alpha}) \prec_r f(\tilde{\beta})$.

1.2. *Megjegyzés.* A $\prec$ jellel jelöljük az $1 < 2 < 3 \ldots < n < n+1 < \ldots$ rendezést. A $\prec_r$ parciális rendezésben a 0 legyen egy kitüntetett elem, amelyre: $0 \prec_r e$, $(e \in E_{\aleph_0})$.

2. A határérték-logikák egy parciális rendezéséről

Ismert tény, hogy kontinuum sok határérték-logika létezik [9]. A [2] dolgozatban megvizsgáltunk néhány természetesen felvetődő parciális rendezést a határérték-logikák halmazán.

Ezeknek a parciális rendezéseknek a segítségével — amelyek különböző homomorfizmusokon alapultak — ekvivalencia relációkat definiáltunk, amelyek a határérték-logikák halmazát felosztják ekvivalens osztályokra. Bebizonyítást nyert, hogy ezeknek a parciális rendezéseknek nincs sem minimális, sem pedig maximális eleme.

Parciális rendezéseket nem csak algebrai úton szokás definiálni, hanem logikai szempontból is. A legnyilvánvalóbb logikai fogalom, a k-értékű logikák modelezhetősége a határérték-logikában. Nevezetesen arról van szó, hogy pontosan meg kell állapítani hol, milyen értékeken valósul meg egy adott k-értékű logika modelezhetősége, és ezek alapján össze lehet hasonlítani a határérték-logikákat. Definiáljuk az M maximális határérték logikát, amelyben minden határérték logika „benne van", és a dolgozat többi paragrafusában ezt az M logikát fogjuk tanulmányozni.

Legyen P határérték-logika, ε az $E_{\aleph_0}$ részhalmaza és $g(x_1, x_2, \ldots, x_n) \in P$.

2.1. DEFINÍCIÓ. Jelölje $g_\varepsilon(x_1, x_2, \ldots, x_n)$ azt a függvényt, amely csak az $\varepsilon \times \varepsilon \times \ldots \times \varepsilon$ halmazon van értelmezve, és ott egyenlő a $g(x_1, x_2, \ldots, x_n)$ függvénnyel. A $g_\varepsilon(x_1, x_2, \ldots, x_n)$ függvényt a $g(x_1, x_2, \ldots, x_n)$ függvény szűkítésének nevezzük az ε halmazra vonatkozóan.

2.2. DEFINÍCIÓ. Azt mondjuk, hogy az $A_k (A_k \subset P_{\aleph_0})$ függvényhalmaz a k-értékű logika modellje az $\varepsilon_k = \{e_0, e_1, \ldots, e_{k-1}\}$ $(k \geqq 2)$ halmazon $(\varepsilon_k \subset E_{\aleph_0})$, ha az $[A_k]$ halmazban létezik olyan $f(x_1, x_2)$ függvény, hogy

$$f_{\varepsilon_k}(x_1, x_2) = \begin{cases} e_{i+1}, & \text{ha } (x_1, x_2) \in \varepsilon_k \times \varepsilon_k \text{ és } \max(x_1, x_2) = e_i, \text{ ahol } 0 \leqq i \leqq k-2; \\ e_0, & \text{ha } (x_1, x_2) \in \varepsilon_k \times \varepsilon_k \text{ és } \max(x_1, x_2) = e_{k-1}, \ (e_0 \prec e_1 \prec \ldots \prec e_{k-1}). \end{cases}$$

Könnyű belátni, hogyha A_k a k-értékű logika modellje az $E_{\aleph_0}$ halmazon, akkor az A_k függvényhalmazban van legalább egy olyan függvényhalmaz, amelynek a szűkítése az ε_k halmazra izomorf a k-értékű logikával.

Valóban vegyük pl. az $f(x_1, x_2)$ függvényt. Nyilvánvaló, hogy az $f_{\varepsilon_k}(x_1, x_2) \leftrightarrow$ $\leftrightarrow W_k(z_1, z_2)$ biztosítja nekünk a kívánt izomorfizmust, ahol $W_k(z_1, z_2)$ *Webb függvény* [6], vagyis

$$W_k(z_1, z_2) = \begin{cases} e, & \text{ha } (z_1, z_2) \in E_k \times E_k \text{ és } \max(z_1, z_2) = e-1, \text{ ahol } e \neq k-1; \\ 0, & \text{ha } (z_1, z_2) \in E_k \times E_k \text{ és } \max(z_1, z_2) = k-1; \end{cases}$$

$(E_k = \{0, 1, \ldots, k-1\}, k \geqq 2)$.

Legyen $A \subseteqq P_{\aleph_0}$.

2.3. DEFINÍCIÓ. A véges ε_k részhalmazok rendszerét az A függvényhalmaz tartományának nevezzük, és T_A-val jelöljük. Az $\varepsilon_k = \{e_0, e_1, \ldots, e_{k-1}\}$ halmaz akkor és csak akkor tartozik a T_A tartományhoz, ha A a k-értékű logika modellje az ε_k halmazon. Vezessük be a következő jelölést

$$E^A_{\aleph_0} = \bigcup_{\varepsilon_k \in T_A} \varepsilon_k.$$

Könnyű belátni, hogy minden A függvényosztályra egy és csak egy T_A tartomány létezik.

Ebben a paragrafusban kimondunk néhány lemmát és tételt minden bizonyítás nélkül. Ezeknek az állításoknak a bizonyítását a [8] dolgozatban megtalálhatjuk.

2.1. LEMMA. Az A függvényosztály akkor és csak akkor határérték-logika, ha a T_A tartománya tartalmaz tetszőleges véges számosságú ε_k halmazt.

2.4. DEFINÍCIÓ. Nevezzük a P határérték-logikát növekvőnek, ha a T_P tartománya tartalmaz olyan véges halmazok végtelen sorozatát $(\Pi = \{\varepsilon_2, \varepsilon_3, \ldots, \varepsilon_i, \ldots\})$, amelyre igaz, hogy $\varepsilon_i \subset \varepsilon_{i+1}$ $(i=2, 3, 4, \ldots)$.

A P határérték-logikát nem metszőnek nevezzük, ha a T_P tartománya olyan $\varepsilon^1_{n_1}, \varepsilon^2_{n_2}, \ldots, \varepsilon^i_{n_i}, \ldots$ véges halmazok végtelen sorozatából áll, amelyekre igaz, hogy

1. $\forall i, j\ (i \neq j,\ i, j = 1, 2 \ldots): \varepsilon^i_{n_i} \cap \varepsilon^j_{n_j} = \emptyset$;

2. ha $\varepsilon \in T_p$, akkor $\exists_i: \varepsilon \subseteqq \varepsilon^i_{n_i}$.

2.5. DEFINÍCIÓ. A B függvényosztály nagyobb az A függvényosztálynál, ha létezik az $E^A_{\aleph_0}$ halmaz olyan δ egyértelmű leképzése az $E^B_{\aleph_0}$ halmazba, amelyre igaz, hogy

1. ha $e_i \neq e_j (e_i, e_j \in E^A_{\aleph_0})$, akkor $\delta(e_i) \neq \delta(e_j)$,

2. ha $\varepsilon \in T_A$, akkor $\delta(\varepsilon) \in T_B$.

Azt a tényt, hogy B nagyobb, mint A így jelöljük: $B \geqq A$. Nevezzük az A és B osztályt hasonlónak, ha $A \geqq B$ és $B \geqq A (A = B)$.

2.6. DEFINÍCIÓ. Minimális határérték-logikának nevezzük azt a logikát, amely kisebb minden határérték-logikánál. Maximális határérték-logikának nevezzük azt a logikát, amely minden határérték-logikánál nagyobb. Azaz a maximális hatáérték-logikában minden határérték-logika benne van.

2.1. TÉTEL. A határérték-logika maximális akkor és csak akkor, ha növekvő. A határérték-logika minimális akkor és csak akkor, ha nem metsző.

2.2. TÉTEL. A maximális (minimális) határérték-logikák hasonlóak egymással.

2.1. *Megjegyzés.* A szakirodalomban eddig csak maximális, ill. minimális határérték-logikákat vizsgáltak, ha csak konkrét logikákat akartak vizsgálni. Ez a tény is arra enged következtetni, hogy az így definiált parciális rendezés (ekvivalencia reláció) a „legtermészetesebb".

Határozzuk meg a maximális határérték-logikák egy M reprezentánsát, amelyet a továbbiakban tanulmányozni fogunk. A továbbiakban ε_k halmaz legyen mindig az $\{1, 2, \ldots, k\}$ halmaz $(k \geqq 2)$.

Definiáljuk a $\mu_k(x_1, x_2) \in P_{\aleph_0} (k \geqq 2)$ függvényt a következőképpen:

$$\mu_k(x_1, x_2) = \begin{cases} e, & \text{ha } (x_1, x_2) \in \varepsilon_k \times \varepsilon_k \text{ és } \max(x_1, x_2) = e-1, \\ & \text{ahol } 1 \leqq e-1 \leqq k-1; \\ 1, & \text{ha } (x_1, x_2) \in \varepsilon_k \times \varepsilon_k \text{ és } \max(x_1, x_2) = k; \\ 0, & \text{különben.} \end{cases}$$

Jelölje M_k a $[\{\mu_k(x_1, x_2)\}]$ függvényhalmazt és $M = \left[\bigcup_{k=2}^{\infty} M_k\right]$.

2.2. LEMMA. $M_k \simeq P_k$.

2.3. TÉTEL. M maximális határérték-logika.

2.4. TÉTEL. Az M maximális határérték-logikában minden bázis végtelen.

Bizonyítás. A tétel közvetlenül adódik abból a tényből, hogy

a) az M_k minden függvénye legfeljebb $k+1$ értéket vesz fel;

b) az M_{k_1} és M_{k_2} által generált függvények legfeljebb k értéket vesznek fel, ahol $k=\max(k_1, k_2)$;

c) az M határérték-logikában minden k ($k \geqq 0$) értékhez létezik olyan $f_k(x_1, \ldots, x_n)$ függvény, amely k értéket vesz fel. A tételt bebizonyítottuk.

3. Az M határérték-logika partíció tartó függvényosztályai

Ebben a részben az M határérték-logika partíció-tartó osztályait vizsgáljuk. Bebizonyítjuk, hogy az M partíció-tartó osztályai a P_k partíció-tartó osztályaihoz hasonlóak, és a számosságuk kontinuum. Sőt, az M határérték-logikában kontinuum sok partíció-tartó majdnem teljes osztály van. Dolgozatunkban a partíció-tartás fogalmát a 0 érték specifikálja.

Legyen $(\varepsilon_0=\{0\}) \cup \varepsilon_{n_1}^{1'} \cup \varepsilon_{n_2}^{2'} \cup \ldots \cup \varepsilon_{n_i}^{i'} \cup \ldots$ az $E_{\aleph_0}$ halmaz páronként diszjunkt részhalmazokra való felbontása. Képezzük ebből az $\varepsilon_0 \cup \varepsilon_{n_1}^{1} \cup \varepsilon_{n_2}^{2} \cup \ldots \cup \varepsilon_{n_i}^{i} \cup \ldots$ felbontását, ahol $\forall i$ $(i=1, 2, \ldots)$; $\varepsilon_{n_i}^{i}=\varepsilon_{n_i}^{i'} \cup \varepsilon_0$. R jelölje ezt a felbontást, és nevezzük kvázipartíciónak. Minthogy az M határérték-logika esetében a partíció-tartó osztály nem lesz zárt, így dolgozatunk további részében a kvázipartíció-tartó kifejezés helyett, röviden csak a partíció-tartó kifejezést használjuk.

3.1. DEFINÍCIÓ. Azt mondjuk, hogy α ekvivalens a β-val az R partíció szerint $(\alpha \sim \beta (\text{mod } R))$, ha α és β a partíció azonos osztályába tartozik. Azt mondjuk, hogy $\tilde{\alpha}=(\alpha_1, \alpha_2, \ldots, \alpha_n)$ ekvivalens $\tilde{\beta}=(\beta_1, \beta_2, \ldots, \beta_n)$ szám n-essel az R partíció szerint, ha $\forall_i$ $(i=1, 2, \ldots, n): \alpha_i \sim \beta_i$ (mod R). Azt mondjuk, hogy az $f(x_1, x_2, \ldots, \ldots, x_n) \in M$ függvény R partíció-tartó, ha $\tilde{\alpha} \sim \tilde{\beta}$ (mod R) esetén igaz, hogy $f(\tilde{\alpha}) \sim \sim f(\tilde{\beta})$ (mod R). U^R jelölje az R partíció-tartó függvények halmazát.

3.1. *Megjegyzés.* Az M határérték-logika függvényei rendelkeznek azzal a tulajdonsággal, hogy $f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)=0$. Azoknak a lemmáknak a bizonyítása, amelyek ebben a részben bizonyítás nélkül szerepelnek bizonyítással együtt megtalálhatók az [1, 3, 10] dolgozatokban.

3.2. *Megjegyzés.* Az U^R típusú majdnem teljes osztályok vizsgálatakor nehézséget okoz, hogy az R számossága nem feltétlenül véges. Ezt a problémát legegyszerűbben úgy tudjuk kikerülni, hogy az R partíció helyett annak az R_{ε_k} szűkítéséről beszélünk, amelyben csak a k-nál kisebb elemek osztályokra való felbontását vizsgáljuk.

3.1. LEMMA. U^R zártosztály.

Nyilvánvaló, hogy ha R egyetlen osztályból áll, ill. az R partíció osztályai a 0 elemen kívül csak egy elemet tartalmaznak, akkor az R partíciónak megfelelő partíció-tartó osztály az M határérték-logikával egyenlő.

Ezt a két speciális partíciót triviálisnak nevezzük.

3.2. LEMMA. Ha R nem triviális partíció, akkor U^R zártosztály nem egyenlő az M határérték-logikával.

Bizonyítás. Mivel R nem triviális partició, úgy létezik olyan $\varepsilon_{n_i}^i$ halmaz, amelyben legalább két, nullától különböző elem van, és $\varepsilon_{n_i}^i \neq E_{\aleph_0}$. Legyen ez a két elem e_1 és e_2. Ezenkívül létezik még az $\varepsilon_{n_i}^i$ osztályon kívüli más osztály is. Legyen e ennek az osztálynak a 0 elemtől különböző eleme.

Vezessük be a következő jelölést

$$g(x) = \begin{cases} e_1, & \text{ha} \quad x = e_1; \\ e, & \text{ha} \quad x = e_2; \\ 0, & \text{különben.} \end{cases}$$

Nyilvánvaló, hogy $g(x) \overline{\in} U^R$. A lemmát bebizonyítottuk.

Az R partíció legyen az $E_{\aleph_0} \setminus 0$ egy diszjunkt felbontása. R_{ε_k} legyen az előbbi R partíció ε_k halmazra való szűkítése. Ennek megfelelően $U_{\varepsilon_k}^R$ legyen mindazon $f(x_1, x_2, \ldots, x_n)$ függvények halmaza az U^R függvényosztályból, amelyekre igaz, hogy $f(x_1, x_2, \ldots, e, \ldots, x_n) = 0$, ha $e \overline{\in} \varepsilon_k$.

3.3. *Megjegyzés.* Ebben a paragrafusban természetszerűen nem az U^R függvényosztályról, hanem annak egy ε_γ halmazra való $U_{\varepsilon_\gamma}^R$ megszorításáról beszélünk (ez vonatkozik az $f(x_1, \ldots, x_n)$ függvényre is). Ez nem megszorítás a tételekre, ill. lemmákra. Ezt az ε_γ halmazt egy $f(x_1, x_2, \ldots, x_n) \in M$ függvényből kiindulva a következőképpen kapjuk meg:

1. jelölje α az $f(x_1, x_2, \ldots, x_n)$ függvény által felvett értékek maximumát;
2. legyen β az a legnagyobb érték, amely valamely szám n-esben előfordulva még nem ad nullát $(f(\ldots, \beta, \ldots) \neq 0)$;
3. $\varepsilon_\gamma = \{0, 1, 2, \ldots, (\gamma = \max(\alpha, \beta))\}$.

3.3. LEMMA. Legyen $f(x_1, x_2, \ldots, x_n) \overline{\in} U^R$. Minden $f(x_1, x_2, \ldots, x_n)$ függvényhez létezik olyan $g_i(x) \in U^R$ függvény $(i = 1, 2, \ldots, n)$, hogy $f(g_1(x), g_2(x), \ldots$ $\ldots, g_n(x)) = g(x) \overline{\in} U^R$.

Bizonyítás. Ha $f(x_1, x_2, \ldots, x_n) \overline{\in} U^R$, akkor létezik olyan $\tilde{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ és $\tilde{\beta} = (\beta_1, \beta_2, \ldots, \beta_n)$, hogy $\tilde{\alpha} \sim \tilde{\beta} \pmod R$ és $f(\tilde{\alpha}) \sim f(\tilde{\beta})$ nem teljesül. Nyilvánvaló, hogy $0 \neq f(\tilde{\alpha}) \neq f(\tilde{\beta}) \neq 0$.

Legyen ε_γ' az ε_γ halmaz egy tetszőleges legalább két elemet tartalmazó osztálya az R_{ε_γ} partíció szerint.

Határozzuk meg a $g_i(x)$ függvényt $(1 \leqq i \leqq n)$ a következőképpen:

$$g_i(x) = \begin{cases} \alpha_i, & \text{ha} \quad x = \gamma_1, \quad \text{ahol} \quad \gamma_1 \in \varepsilon_\gamma'; \\ \beta_i & \text{ha} \quad x = \gamma_2 \neq \gamma_1 \quad \text{és} \quad \gamma_2 \in \varepsilon_\gamma'; \\ 0, & \text{különben.} \end{cases}$$

Nyilvánvaló, hogy az $f(g_1(x), g_2(x), \ldots, g_n(x)) = g(x)$ függvény rendelkezik a kívánt tulajdonsággal, mivel

$$g(x) = \begin{cases} f(\tilde{\alpha}), & \text{ha} \quad x = \gamma_1, \\ f(\tilde{\beta}), & \text{ha} \quad x \neq \gamma_1 \quad \text{és} \quad x \in \varepsilon_\gamma'; \\ 0, & \text{különben.} \end{cases}$$

A lemmát bebizonyítottuk.

3.1. TÉTEL. Az U^R függvényosztály majdnem teljes az M határérték-logikában, ha R nem triviális partíció.

Bizonyítás. Elegendő bebizonyítani, hogy $[\{f(x_1, x_2, ..., x_n)\} \cup U^R] = M$, $(f(x_1, x_2, ..., x_n) \overline{\in} U^R)$. Legyen $k \geqq \gamma$, ahol a γ elemet 3.3. megjegyzés alapján az $f(x_1, x_2, ..., x_n)$ függvény segítségével határozzuk meg. Mutassuk meg, hogy az $f(x_1, x_2, ..., x_n)$ függvény és az $U^R_{\varepsilon_k}$ halmaz generálja a $[\{\mu_k(x_1, x_2)\}] = M_k$ logikát, amely izomorf a P_k k-értékű logikával.

Vezessük be a következő jelöléseket:

$$R_{\varepsilon_k} = \{\varepsilon_0 = \{0\}, \varepsilon^1, \varepsilon^2, ..., \varepsilon^t\} \quad \text{és} \quad \varepsilon^i = \{a^i_1, a^i_2, ..., a^i_{n_i}\} \quad (1 \leqq i \leqq t).$$

Határozzuk meg az $s(x)$ függvényt a következőképpen:

$$s(x) = \begin{pmatrix} 0 & 1 & 2 \ldots n_1 n_1 + 1 & n_1 + 2 \ldots n_1 + n_2 n_1 + n_2 + 1 \ldots n_1 + n_2 + \ldots + n_t \\ 0 & a^1_1 & a^1_2 \ldots a^1_{n_1} \; a^2_1 \quad a^2_2 \quad \ldots \quad a^2_{n_2} & a^3_1 \quad \ldots \quad a^t_{nt}. \end{pmatrix}$$

Nyilvánvaló, hogy az $U^R_{\varepsilon_k}$ osztály az $U^R_{S(\varepsilon_k)}$ osztály duálisa az $s(x)$ függvényhez viszonyítva.

Az 1.1. megjegyzés alapján elég azt bebizonyítani, hogy az $U^R_{S(\varepsilon_k)}$ osztály majdnem teljes az M_k logikában.

A 3.3. lemma segítségével az $f(x_1, x_2, ..., x_n) \overline{\in} U^R$ függvényből kapunk egy $g(x) \overline{\in} U^R$ függvényt $(g(\gamma_1) = f(\tilde{\alpha}) = \alpha,\ g(\gamma_2) = f(\tilde{\beta}) = \beta,\ \alpha < \beta$ és $\alpha \sim \beta$ nem teljesül).

Vezessük be a következő jelölést:

$$h_{\varepsilon_k}(x) = \begin{cases} \gamma_2, & \text{ha} \quad x = \gamma_1; \\ \gamma_1, & \text{ha} \quad x = \gamma_2, \quad \text{ahol} \quad \gamma_1, \gamma_2 \in \varepsilon'_\gamma; \\ 0, & \text{különben.} \end{cases}$$

Mivel γ_1 és γ_2 ugyanabba az ε'_γ osztályba tartozik, azért $h_{\varepsilon_k}(x) \in U^R_{\varepsilon_k}$.

Jelölje $\varphi_{\varepsilon_k}(x)$ a következő függvényt:

$$\varphi_{\varepsilon_k}(x) = \begin{cases} 0, & \text{ha} \quad x = 0; \\ 1, & \text{ha} \quad x \in \varepsilon_k \quad \text{és} \quad x \overline{\in} \varepsilon'_\gamma; \\ k, & \text{ha} \quad x \in \varepsilon_k \quad \text{és} \quad x \in \varepsilon'_\gamma. \end{cases}$$

Nyilvánvaló, hogy $\varphi_{\varepsilon_k}(x) \in U^R_{\varepsilon_k}$, továbbá vezessük be a következő függvényt:

$$\chi^i_{\varepsilon_k}(x) = \begin{cases} 0, & \text{ha} \quad x = 0; \\ \gamma_1, & \text{ha} \quad 1 \leqq x < i; \\ \gamma_2, & \text{ha} \quad k \geqq x \geqq i. \end{cases}$$

Ekkor, $\chi^i_{\varepsilon_k}(x) \in U^R_{\varepsilon_k}$ $(i = 2, 3, ..., k)$, mivel $\gamma_1, \gamma_2 \in \varepsilon'_\gamma$. Mutassuk meg, hogy a $\max_{\varepsilon_k}(x_1, x_2)$, $\min_{\varepsilon_k}(x_1, x_2)$, $C^j_{\varepsilon_k} = j$ $(j = 1, 2, ..., k)$ és az $m^i(x) = \varphi_{\varepsilon_k}[g\{\chi^i_{\varepsilon_k}(x)\}]$ $(i = 2, ..., k)$ függvények (amelyek az $U^R_{\varepsilon_k}$ osztályhoz tartoznak) generálják az $1 < 2 < ... < k$ rendezés szerinti összes monoton függvényt, ahol

$$m^i(x) = \begin{cases} 0, & \text{ha} \quad x = 0; \\ 1, & \text{ha} \quad 1 \leqq x < i; \\ k, & \text{ha} \quad k \geqq x \geqq i. \end{cases}$$

Vezessük be a következő jelölést:

$$Z_{\varepsilon_k}^{\tilde{\alpha},\beta}(x_1, x_2, \ldots, x_n) = \min_{\varepsilon_k}(\beta, m^{\alpha_1}(x_1), m^{\alpha_2}(x_2), \ldots, m^{\alpha_n}(x_n)),$$

ahol $\tilde{\alpha}=(\alpha_1, \alpha_2, \ldots, \alpha_n)$; $1 \leqq \alpha_t \leqq k$; $t=1, 2, \ldots, n$ és $m^1(x)=k$. Nyilvánvaló, hogy

$$Z_{\varepsilon_k}^{\tilde{\alpha},\beta}(x_1, \ldots, x_n) = \begin{cases} \beta, & \text{ha } \tilde{x} \geqq \tilde{\alpha}; \\ 1, & \text{különben.} \end{cases}$$

Minden monoton függvényre igaz, hogy

$$Z(x_1, x_2, \ldots, x_n) = \max_{\tilde{\alpha}} \{Z^{\tilde{\alpha}}, Z^{\tilde{\alpha},f(\tilde{\alpha})}(x_1, x_2, \ldots, x_k)\}.$$

Az egyváltozós monoton függvények $M_{\varepsilon_k}^1$, $h_{\varepsilon_k}(x)$ és a $g(x)$ függvény generálja a $h(x)$ függvényt, ahol

$$h(x) = \begin{cases} \beta, & \text{ha } 1 \leqq x \leqq k-1; \\ \alpha, & \text{ha } x = k; \\ 0, & \text{különben.} \end{cases}$$

Definiálja a $t(x, y)$ monoton függvényt a következő értéktáblázat:

y \ x	0	1	2	...	α	...	$\beta-1$	β	$\beta+1$	...	$k-1$	k	$k+1$	...
0	0	0	0	...	0	...	0	0	0	...	0	0	0	...
1	0	1	1	...	1	...	1	2	2	...	2	2	0	...
2	0	1	1	...	1	...	1	3	3	...	3	3	0	...
⋮	⋮	⋮	⋮	...	⋮	...	⋮	⋮	⋮	...	⋮	⋮	⋮	...
α	0	1	1	...	1	...	1	$\alpha+1$	$\alpha+1$	...	$\alpha+1$	$\alpha+1$	0	...
⋮	⋮	⋮	⋮	...	⋮	...	⋮	⋮	⋮	...	⋮	⋮	⋮	...
$\beta-1$	0	1	1	...	1	...	1	β	β	...	β	β	0	...
β	0	1	1	...	1	...	1	$\beta+1$	$\beta+1$	...	$\beta+1$	$\beta+1$	0	...
$\beta+1$	0	1	1	...	1	...	1	$\beta+2$	$\beta+2$	...	$\beta+2$	$\beta+2$	0	...
⋮	⋮	⋮	⋮	...	⋮	...	⋮	⋮	⋮	...	⋮	⋮	⋮	...
$k-1$	0	1	1	...	1	...	1	k	k	...	k	k	0	...
k	0	1	1	...	1	...	1	k	k	...	k	k	0	...
$k+1$	0	0	0	...	0	...	0	0	0	...	0	0	0	...
⋮	⋮	⋮	⋮	...	⋮	...	⋮	⋮	⋮	...	⋮	⋮	⋮	...

Könnyű belátni, hogy $\max_{\varepsilon_k}(t(h(x), x), t(h(y), y)) = \mu_k(x, y)$. A tételt bebizonyítottuk.

3.2. TÉTEL. Az M határérték-logikában kontinuum sok majdnem teljes osztály van.

Bizonyítás. Az állítás következik abból a tényből, hogy a különböző R partícióknak megfelelő U^R majdnem teljes osztályok száma is kontinuum. Mivel az M megszámlálható sok függvényből áll, így több mint kontinuum sok majdnem teljes

osztály az M-ben nem is lehet. Az, hogy az U^R osztályok száma kontinuum, következik abból, hogy

a) a különböző R partíciók száma kontinuum;

b) különböző R-eknek, különböző U^R felel meg.

A tételt bebizonyítottuk.

4. Az M határérték-logikában levő majdnem teljes monoton és önduális osztályok

4.1. DEFINÍCIÓ. Legyen $\prec_r$ az $E_{\aleph_0} \setminus 0$ halmaz olyan parciális rendezése, hogy tetszőleges ε_k halmazra $(k \geqq 2)$ létezik olyan α és β, hogy $\alpha \prec_r e \prec_r \beta$ minden $e \in \varepsilon_k$ esetén. Nevezzük ezeket a parciális rendezéseket monoton parciális rendezésnek.

4.1. TÉTEL. Egy monoton zártosztály majdnem teljes az M határérték-logikában akkor és csak akkor, ha a $\prec_r$ parciális rendezése monoton.

4.2. TÉTEL. A monoton majdnem teljes osztályok száma kontinuum.

4.3. TÉTEL. Egy önduális osztály majdnem teljes az M határérték-logikában akkor és csak akkor, ha az $s(x)$ függvény prímrendű.

4.4. TÉTEL. Az önduális majdnem teljes osztályok száma kontinuum.

4.5. TÉTEL. Minden monoton, ill. önduális zárt osztály kiterjeszthető az M határérték-logikában majdnem teljes monoton, ill. önduális zárt osztályig.

A 4.1—4.5. tételek az [1, 3, 4, 10] dolgozatok felhasználásával analóg módon bizonyíthatók, mint a 3. szakasz tételei. A 4.5. tétel érdekességét az a tény adja, hogy eltérően a P_k véges, és $P_{\aleph_0}$ végtelenértékű logikától [7, 10], az M határérték-logikában nem minden zárt osztály terjeszthető ki majdnem teljes osztályig. Ezért az M határérték-logikában a *Kuznyecov tétel* sem érvényes [10].

4.6. TÉTEL. Az M határérték-logikában nem minden zártosztály terjeszthető ki majdnem teljes osztállyá.

Bizonyítás. Valóban, vegyük az M összes egyváltozós függvényeit. Ez zárt osztály lesz. Ha ehhez bármilyen véges számú több változós függvényt hozzáveszünk, nem kaphatjuk meg az M maximális határérték-logikát, mivel legfeljebb csak azokat az M_k-kat kaphatjuk meg, ahol a k nem nagyobb annál az értéknél, amit a többváltozós függvényünk felvett.

A tételt bebizonyítottuk.

IRODALOM

[1] BAGYINSZKI, J., „Az m-értékű logika függvényrendszereinek funkcionális teljessége", *KFKI-73-55* (1973) 1—111.

[2] DEMETROVICS, J., „A határérték-logikák homomorfizmusáról", *Alk. Mat. Lapok* **1** (1975) 125—138.

[3] ROSENBERG, I., »La structure des funktions de plusieurs variables sur un ensemble fini«, *C. r. Acad. Sci. Paris* **14** (1969) 413—438.

[4] ROSENBERG, I., »Über die Verschidenkeit maximaler Klassen«, *Rev. Roum. math. jures et appl.* **14** (1969) 413—438.

[5] SALOMAA, A., "Some completeness criteria for sets of functions over a finite domain I, II", *Annales Universitates Turkuensis* **53** (1962) 1—9 and **63** (1963) 1—19.
[6] WEBB, P., "Generation of any n-valued logic by are binary operator", *Proc. Nat. Acad. Sci.* **21** (1935) 252—254.
[7] Гаврилов, Г. П., «О мощности множества предельных логик, обладающих конечным базисом», *Сб. проблемы кибернетики* **21** (1969) 115—126.
[8] Деметрович, Я., «О некоторых гомоморфизмах и отношениях для предельных логик», *Сб. проблемы кибернетики* **30** (1975) 5—42.
[9] Деметрович, Я., «О числе попарно неизоморфных предельных логик», *Дискретный анализ* **24** (1974) 21—29.
[10] Яблонский, С. В., «Функциональные построения в K-значной логике», *Труды мат. института им. В. А. Стеклова А. Н. СССР* **51** (1958) 5—142.

(Beérkezett: 1976. május 16.)

DEMETROVICS JÁNOS
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1132 BUDAPEST XIII., VICTOR HUGO U. 18—22.

THE M MAXIMAL LIMIT — LOGIC

J. DEMETROVICS

In the present paper we prove, that the set of limit logics has a decomposition to classes possessing those of the maximal and the minimal ones. In detail a representative of the maximal limit logics M is investigated.

We give a sufficient and necesseary condition for a class in the limit logics M to be almost complete in the cases if it is partition — preserving, selfdual or monotone. We prove that in any of the above three cases the cardinality of almost complete classes is continuum. It is shown, that for M the theorem of KOOZNIETZOV doesn't hold.

A closed class is constructed, which cannot be extended to an almost complete one.

Alkalmazott Matematikai Lapok **2** *(1976) 67—81*

A GEOMETRIAI PROGRAMOZÁS EGY MEGOLDÁSI MÓDSZERE[1]

KÁDAS SÁNDOR
Budapest

A geometriai programozási duál feladat egy konkáv függvény maximalizálása lineáris egyenlőség-feltételek mellett. Természetes gondolat ennek a megoldására egy gradiens vetítési módszert alkalmazni, azonban ezt a feladat speciális követelményeihez kell idomítani. Egy ilyen, a gradiens vetítési módszerre támaszkodó, heurisztikus elemeket is tartalmazó algoritmus és a vele kapcsolatos számítástechnikai tapasztalatok leírása szerepel a dolgozatban. Az első részben található a geometriai programozási feladat megfogalmazása és dualitási tételének ismertetése, a másodikban a tárgyalt algoritmus váza, a harmadikban a geometriai programozási feladat két egyszerű, de érdekes és az algoritmus szempontjából hasznos tulajdonságának bizonyítása, míg a befejező rész tartalmazza az algoritmus részletesebb leírását, a nyert számítástechnikai tapasztalatok ismertetését.

1. A geometriai programozási feladat megfogalmazása és dualitási elmélete

A geometriai-, vagy más néven pozinomiális-programozás alapfeladata egy pozitív együtthatós általánosított polinom (ezt nevezik pozinomnak) minimalizálása pozinomiális korlátozó feltételek mellett. Ehhez hozzá lehet rendelni egy lineárisan korlátozott duál feladatot, a kapott primál-duál feladatpár a következő:

Primál feladat:

$$(P)\begin{cases}\min g_0(\mathbf{t}), \\ \text{feltéve, hogy} \\ \mathbf{t} = (t_1, \ldots, t_m) > \mathbf{0}, \\ g_1(\mathbf{t}) \leqq 1, \\ \vdots \\ g_p(\mathbf{t}) \leqq 1, \end{cases}$$

ahol $g_k(\mathbf{t}) = \sum_{i \in J[k]} C_i t_1^{a_{i1}} \ldots t_m^{a_{im}}, \quad C_i > 0,$

a_{ij} tetszőleges valós számok

$$J[k] = \{m_k, m_k+1, \ldots, n_k\} \quad k = 0, 1, \ldots, p,$$

$$m_0 = 1, \; n_p = n, \; m_k = n_{k-1}+1,$$

tehát összesen m primál változó és n primál additív tag van.

[1] A szerző ezúton mond köszönetet DR. GERENCSÉR LÁSZLÓ és DR. KLAFSZKY EMIL lektoroknak a dolgozat végleges formája kialakításában nyújtott segítségükért.

Duál feladat:

$$(D)\begin{cases} \max v(\mathbf{y}) = \prod_{i=1}^{n}\left(\frac{C_i}{y_i}\right)^{y_i}\prod_{k=1}^{P}\lambda_k(\mathbf{y})^{\lambda_k(\mathbf{y})}, \\ \text{ahol } \lambda_k(\mathbf{y}) = \sum_{i\in J[k]} y_i, \\ \text{feltéve, hogy } \mathbf{y} \geqq \mathbf{0}, \\ \sum_{i\in J[0]} y_i = 1 \text{ („normalitási” feltétel),} \\ \sum_{i=1}^{n} a_{ij}y_i = 0, \quad j = 1, \ldots, m \text{ („ortogonalitási” feltételek),} \end{cases}$$

továbbá definícóiszerűen $v(\mathbf{y})$-ban $x^x=1$ $x=0$ esetén.

A feladat ilyen megfogalmazása R. DUFFINtól (1. pl. [4]) származik. Egy másik, szimmetrikusabb, de kevésbé használt alak is létezik, ez pl KLAFSZKY EMIL [8] munkájában található meg. Látható, hogy a duálfeladat változóinak száma (n) azonos a primál additív kifejezések számával, minden primál additív taghoz hozzárendelhető egy duálváltozó, s a k-adik primál feltételnek megfelelő duálváltozók összege ādja $\lambda_k(\mathbf{y})$-t. A $d=n-m-1$ mennyiséget (a 3.1. lemma utáni 1. megjegyzésben látjuk majd, hogy $n>m$ feltehető, s így $d>0$), amely az $\mathbf{y}$ „szabadsági foka”-ként interpretálható, a feladat bonyolultsági fokának (“degree of difficulty”) szokták nevezni. Exponens mátrixnak nevezzük az

$$\mathbf{A}' = \begin{pmatrix} a_{11} & a_{21} & \ldots & a_{n1} \\ \vdots & \vdots & & \vdots \\ a_{1m} & a_{2m} & \ldots & a_{nm} \end{pmatrix} \text{ mátrixot,}$$

tehát a duál ortogonalitási feltételek összefoglalva:

$$\mathbf{A}'\mathbf{y} = \mathbf{0}.$$

($\mathbf{A}'$ itt és a továbbiakban az $\mathbf{A}$ mátrix transzponáltját jelenti.) A primál feladat helyett néha az $e^{z_i}=t_i$, $i=1, \ldots, m$ helyettesítésekkel kapott (P_z) transzformált primál feladatot vizsgáljuk. A duál célfüggvényt a logaritmusával helyettesítve a logaritmusfüggvény monotonitása miatt az eredetivel ekvivalens feladatot kapunk, ezt jelöljük (D')-vel, tehát ennek a célfüggvénye $c_i=\log C_i$ jelöléssel:

$$\log v(\mathbf{y}) = \sum_{i\in J[0]} y_i(c_i - \log y_i) + \sum_{k=1}^{P}\sum_{i\in J[k]} y_i\left(c_i + \log\frac{\lambda_k(\mathbf{y})}{y_i}\right). \tag{1.1}$$

Most a geometriai programozás dualitáselméletének néhány, alapvető, a bemutatásra kerülő algoritmus szempontjából is fontos eredményét ismertetjük. Bizonyításuk megtalálható a [4] és a [8] munkákban.

A dualitáselméletben alapvető a geometriai programozás ún. „főlemmája” (az elnevezés DUFFIN-tól származik):

1.1. TÉTEL. Ha $\mathbf{t}$ a (P) primál, $\mathbf{y}$ a (D) duál feladat megengedett megoldása, akkor

$$g_0(\mathbf{t}) \geqq v(\mathbf{y}). \tag{1.2}$$

Továbbá, azonos feltételek mellett $g_0(\mathbf{t})=v(\mathbf{y})$ akkor és csak akkor, ha

$$y_i = \begin{cases} \dfrac{C_i t_1^{a_{i1}} t_2^{a_{i2}} \dots t_m^{a_{im}}}{g_0(\mathbf{t})}, & i \in J[0] \\ \lambda_k(\mathbf{y}) C_i t_1^{a_{i1}} t_2^{a_{i2}} \dots t_m^{a_{im}}, & i \in J[k],\ k = 1, \dots, p. \end{cases} \tag{1.3}$$

Ennek a bizonyítása a geometriai egyenlőtlenségre támaszkodik (innen ered a feladatkör elnevezése), amely a következő. Ha x_i, y_i $(i=1, 2, \dots, m)$ tetszőleges nemnegatív számok, akkor:

$$\left(\frac{\sum_{i=1}^{m} x_i}{\sum_{i=1}^{m} y_i} \right)^{\sum_{i=1}^{m} y_i} \geqq \prod_{i=1}^{m} \left(\frac{x_i}{y_i} \right)^{y_i}, \tag{1.4}$$

ahol $\left(\frac{x}{y}\right)^0 = 1$ definíció szerint $x=0$ vagy $y=0$ esetén is. Egyenlőség akkor és csak akkor teljesül, ha

$$x_j \sum_{i=1}^{m} y_i = y_j \sum_{i=1}^{m} x_i, \quad j = 1, 2, \dots, m.$$

A (P) primál feladatot szuperkonzisztensnek nevezzük, ha van olyan $\mathbf{t}>\mathbf{0}$ vektor, amely határozott egyenlőtlenséggel elégíti ki mindegyik primál feltételt. A dualitásra vonatkozik a következő tétel:

1.2. TÉTEL. Tegyük fel, hogy a (P) feladat szuperkonzisztens és hogy a $g_0(\mathbf{t})$ primál függvény felveszi a primál megengedett halmazra vonatkozó infimumát. Ekkor

1. a megfelelő (D) duálprogram konzisztens és a $v(\mathbf{y})$ duálfüggvény felveszi a duál-megengedett halmazra vonatkozó supremumát,

2. a duálprogram maximumértéke megegyezik a primálprogram minimumértékével,

3. ha $\hat{\mathbf{t}}$ a (P) egy minimalizáló pontja, akkor vannak olyan nem-negatív $\hat{\mu}_k$ $(k=1, \dots, p)$ *Lagrange-együtthatók,* hogy az

$$L(\mathbf{t}, \boldsymbol{\mu}) = g_0(\mathbf{t}) + \sum_{k=1}^{p} \mu_k [g_k(\mathbf{t}) - 1]$$

Lagrange-függvény rendelkezik a következő tulajdonsággal:

$$L(\hat{\mathbf{t}}, \boldsymbol{\mu}) \leqq g_0(\hat{\mathbf{t}}) = L(\hat{\mathbf{t}}, \hat{\boldsymbol{\mu}}) \leqq L(\mathbf{t}, \hat{\boldsymbol{\mu}}) \tag{1.5}$$

bármely $\mathbf{t}>\mathbf{0}, \boldsymbol{\mu}>\mathbf{0}$ esetén. Továbbá van egy olyan $\hat{\mathbf{y}}$ maximalizáló pont a (D)-re, melynek komponensei:

$$\hat{y}_i = \begin{cases} \dfrac{C_i \hat{t}_1^{a_{i1}} \dots \hat{t}_m^{a_{im}}}{g_0(t)}, & i \in J[0] \\ \dfrac{\mu_k C_i \hat{t}_1^{a_{i1}} \dots \hat{t}_m^{a_{im}}}{g_0(\mathbf{t})}, & i \in J[k],\ k = 1, \dots, p \end{cases} \tag{1.6}$$

és itt teljesül:

$$\lambda_k(\hat{\mathbf{y}}) = \frac{\hat{\mu}_k}{g_0(\hat{\mathbf{t}})} \quad k = 1, \dots, p, \tag{1.7}$$

4. véve a (D) duál program egy $\hat{\mathbf{y}}$ maximalizáló pontját, minden $\hat{\mathbf{t}}$ primál minimalizáló pont kielégíti a következő egyenletrendszert, melynek egyenleteit egyensúlyi feltételeknek is szokták nevezni:

$$C_i \hat{t}_1^{a_{i1}} \dots \hat{t}_m^{a_{im}} = \begin{cases} \hat{y}_i v(\hat{\mathbf{y}}), & i \in J[0] \\ \hat{y}_i / \lambda_k(\hat{\mathbf{y}}), & i \in J[k] \quad \text{és} \quad k: \lambda_k(\hat{\mathbf{y}}) > \mathbf{0}. \end{cases} \tag{1.8}$$

Ezt a tételt az elméleti szempontból elég erős szuperkonzisztencia megkötés miatt néha gyenge dualitási tételnek is nevezik. Lényegesen mélyebb állítás a következő (az ún. „erős" dualitási tétel):

1.3. TÉTEL. Ha a (D) feladatnak van $\mathbf{y}>\mathbf{0}$ megengedett megoldása, és a célfüggvénye felülről korlátos, akkor a primál célfüggvény felveszi minimumát a feltételi halmaz valamely $\mathbf{t}^0$ pontjában, és

$$g_0(\mathbf{t}^0) = \sup_{\mathbf{y} \in (D)} v(\mathbf{y}).$$

Megjegyzés. (1.3)-ból látszik, hogy optimális $(\mathbf{t}, \mathbf{y})$ pár esetén a primál célfüggvényhez tartozó y_i duálváltozók pozitívak, a k-adik primál feltételhez tartozó y_i duálváltozók vagy mind 0 értékűek, vagy mind pozitívak, s az utóbbi esetben az említett feltétel aktív:

$$g_k(\mathbf{t}) = 1.$$

Az optimum stabilitását a paraméterek kis változása esetén KLAFSZKY EMIL eredménye ([8], 75. old.) biztosítja. Az $e^{z_i}=t_i$, $i=1, \dots, m$ helyettesítéssel kapott (P_z) transzformált primál feladat célfüggvénye és feltételi függvényei egyaránt konvexek (hiszen ezek exponenciális függvények pozitív együtthatós összegei), tehát (P_z) konvex feladat. Könnyen bizonyítható, hogy az (1.1) logaritmikus duálfüggvény konkáv, tehát a duál feladat egy konkáv függvény maximalizálása lineáris feltételek mellett. Ez a feladat számítástechnikailag jól kezelhető, a fő probléma az, hogy a $v(\mathbf{y})$ duálfüggvény a megengedett tartomány határán nem differenciálható.

Kézenfekvő a (D) duál feladat megoldására valamilyen gradiens módszert alkalmazni. Elsősorban az irodalmi utalások szerint lineáris feltételek esetén jól működő redukált gradiens- és gradiens vetítési módszerek jönnek számításba. Az utóbbi módszer előnyösebbnek látszik, mivel ennél könnyebb megszervezni a változók határtól való „elszigetelését" (erre azért van szükség, mert a $v(\mathbf{y})$ duálfüggvény deriváltja a duál megengedett tartomány határán nem létezik). Ez a módszer az alapja a következőkben leírt algoritmusnak.

2. Egy gradiens vetítési módszer

Ez az algoritmus NIJKAMP [10]-ben közölt algoritmusának továbbfejlesztéséből, konkretizálásából származik. A módszer sok tekintetben heurisztikus (az alapul szolgáló [10]-beli változat is az), a duál feladat speciális problémáinak megoldására és a konvergencia gyorsítására heurisztikus eljárásokat használ. Ezen eljárások alapötlete általában kézenfekvő, de konkrét, hatékonyan működő formájuk kidolgozása sok számítógépes kísérletezést igényel.

Ebben a részben a tárgyalandó algoritmus vázát ismertetjük, a speciális problémák részletes ismertetésére a 4. részben kerül sor.

Az (1.1)-ben megfogalmazott (P) geometriai programozási primál feladatot a hozzárendelt (D) duál feladaton keresztül oldjuk meg, kihasználva annak a lineáris feltételekből adódó előnyös tulajdonságait. Kézenfekvő gondolat a (D) megoldásánál a célfüggvény gradiensének a lineáris egyenlőség feltételekre vett „vetületét" használni haladási irányként az egyes iterációkban. Ennek egy megvalósítását adja NIJKAMP [10]-ben.

Analitikusan kiszámítja a „feltételek mentén" haladó, leggyorsabb növekedést biztosító irányt. Jólismert tény, hogy ez éppen a lineáris egyenlőség feltételek által meghatározott altérre vetített gradiens iránya. Jelen (D) feladat esetében az $\mathbf{y}_0$ megengedett pontban számított $\mathbf{d}$ vetített gradiens a következőképpen kapható meg:

$$\mathbf{d} = \{\mathbf{E} - \mathbf{A}^{*\prime}(\mathbf{A}^*, \mathbf{A}^{*\prime})^{-1}\mathbf{A}^*\} \nabla \log v(\mathbf{y}_0). \tag{2.1}$$

Itt $\mathbf{A}^*$ $\mathbf{A}'$-nek a normalitási feltétel sorával kibővített változatát jelöli. Az itt szereplő $\mathbf{H} = \mathbf{E} - \mathbf{A}^{*\prime}(\mathbf{A}^*\mathbf{A}^{*\prime})^{-1}\mathbf{A}^*$ az ún. vetítési mátrix.

Az algoritmus váza abból áll, hogy egy megengedett megoldásból elindulva iterációnként az előzőleg kapott pontból kiindulva, a (2.1) által meghatározott irány mentén valamilyen módszerrel maximalizáljuk a $\log v(\mathbf{y})$ célfüggvényt az $\mathbf{y} \geqq \mathbf{0}$ feltétel megsértése nélkül. Ha az iterációk során a duál optimális megoldást már elég pontosan kiszámítottuk, akkor az (1.8) egyensúlyi feltételek alapján egy lineáris egyenletrendszer megoldásával kiszámítjuk a hozzá tartozó primál optimális megoldást.

3. A geometriai programozási feladat két tulajdonsága

Az előző részben leírt algoritmusvázlat részletes kidolgozásakor, a felmerült és az elvben még felléphető problémák megoldása során a geometriai programozási feladat néhány érdekesnek látszó, bár igen könnyen bizonyítható tulajdonsága is adódott. Ezeket tartalmazza a most következő rész, melyben az 1. részben már bevezetett jelöléseket használjuk.

3.1. LEMMA. Tegyük fel, hogy a (P), (D) feladatok mindegyikének van optimális megoldása, s hogy $n > m$. Ekkor

a) Ha $\mathbf{A}'$ — az exponens mátrix — bármely m különböző oszlopa lineárisan független, akkor a primál feladat optimális megoldása egyértelmű.

b) Az (1.8)-beli egyenletek logaritmálásával kapott egyensúlyi egyenletek — ill. ezek együtthatóvektorai — lineárisan összefüggők.

Bizonyítás.

a) A dualitási tétel értelmében (a $t_i = e^{z_i}$ helyettesítéssel áttérve a transzformált (P_z) feladatra), minden $(\mathbf{z}, \mathbf{y})$ primál-duál optimális pár kielégíti az (1.8)-ból adódó

$$\sum_{j=1}^{m} a_{ij} z_j = \begin{cases} \log(y_i \cdot v(\mathbf{y})) - c_i, & i \in J[0] \\ \log \dfrac{y_i}{\lambda_k(\mathbf{y})} - c_i, & i \in J[k],\ \lambda_k(\mathbf{y}) \neq 0 \end{cases} \tag{3.1}$$

egyenletrendszert.

Itt $n-m-1$-nél több duálváltozó nem lehet 0, mert ellenkező esetben az $\mathbf{A}'\mathbf{y}=\mathbf{0}$ ortogonalitási feltétel rendszert (mivel ekkor a nem nulla y_i-hez tartozó $\mathbf{A}$-beli oszlopok a feltevés miatt lineárisan függetlenek) csak az $\mathbf{y}=\mathbf{0}$ elégítené ki, tehát (D) nem lenne konzisztens.

Így a (3.1) rendszerben az egyenletek száma az ismeretlenekét (amelyek a $\mathbf{z}$ komponensei) meghaladja, s az együtthatómátrix bármely négyzetes része nemszinguláris, tehát a megoldás $\mathbf{z}$-ben egyértelmű, ha létezik, s létezését feltevésünk biztosítja.

b) Az állítás triviális, ha az (1.8), ill. a vele ekvivalens (3.1) rendszer egyenleteinek a száma (tehát az optimális duálvektor pozitív komponenseinek a száma) m-nél nagyobb.

Ha ez a szám $\leqq m$, és a kérdéses együttható vektorok (melyek $\mathbf{A}'$ oszlopvektorai) lineárisan függetlenek lennének, akkor tekintve, hogy ezen vektorok és az optimális $\mathbf{y}$ pozitív komponensei között egyértelmű megfeleltetés áll fenn (3.1) miatt, az $\mathbf{A}'\mathbf{y}=\mathbf{0}$ feltételből $\mathbf{y}=\mathbf{0}$ következne, ez viszont ellentmondás.

1. *Megjegyzés.* A lemmában feltételeztük, hogy $n>m$. A következő módon lehet visszavezetni egy olyan feladatot, melyben $m \geqq n$ a szokásos $m<n$ feltételt kielégítő alakra. Ha $m \geqq n$ esetén $\mathbf{A}'$ rangja n, akkor a duál feladat ortogonalitási feltételeit csak az $\mathbf{y}=\mathbf{0}$ triviális megoldás elégíti ki, ez viszont a normalitási feltételnek nem tesz eleget, tehát ekkor a feladat nem konzisztens. Így feltehetjük, hogy $\mathbf{A}'$ rangja $r<n$, s akkor sorai közül kiválasztva egy $r<n$ elemű, bázist alkotó rendszert, a többi sor kifejezhető ezek lineáris kombinációjaként. Ennek következménye, hogy az m számú primál változó $(t_i, i=1, \ldots, m)$ helyett $r<n$ $(\leqq m)$ új változó alkalmazásával is megfogalmazható a primál feladat az eredetivel ekvivalens formában. Ezt az állítást egy példán szemléltetjük. Legyen az eredeti primál feladat:

$$\min (c_1 t_1 t_2^{1,5} t_3^{-1} t_4^3 + c_2 t_2^{-0,5} t_3 t_5),$$

feltéve, hogy

$$\mathbf{t} = (t_1, \ldots, t_5) > \mathbf{0},$$

$$c_3 t_1^{0,5} t_2 t_3^{-2} t_4^{0,5} t_5^{-1,5} + c_4 t_1^{3,5} t_2^{-2} t_3^{2,5} t_4^2 t_5^6 \leqq 1.$$

Az exponens mátrix transzponáltja most

$$\mathbf{A}' = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \\ \mathbf{a}_4 \\ \mathbf{a}_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0,5 & 3,5 \\ 1,5 & -0,5 & 1 & -2 \\ -1 & 1 & -2 & 2,5 \\ 3 & 0 & 0,5 & 2 \\ 0 & 1 & -1,5 & 6 \end{bmatrix}.$$

Itt látható, hogy $\mathbf{A}'$ sorvektorainak $(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)$ egy bázisát alkotja, tehát most $r=3$ $(n=4, m=5)$ és

$$\mathbf{a}_4 = \mathbf{a}_1 + 2\mathbf{a}_2 + \mathbf{a}_3,$$

$$\mathbf{a}_5 = \mathbf{a}_1 + \mathbf{a}_3.$$

Ezt az $e^{z_i}=t_i$ helyettesítéssel kapott (P_z) feladat egy általános tagjára alkalmazva:

$$c_i t_1^{a_{i1}} \dots t_5^{a_{i5}} = c_i \exp(z_1 a_{i1} + \dots + z_5 a_{i5}) =$$

$$= c_i \exp[z_1 a_{i1} + z_2 a_{i2} + z_3 a_{i3} + z_4(a_{i1} + 2a_{i2} + a_{i3}) + z_5(a_{i1} + a_{i3})] =$$

$$= c_i \exp[(z_1 + z_4 + z_5) a_{i1} + (z_2 + 2z_4) a_{i2} + (z_3 + z_5) a_{i3}].$$

Bevezetve a

$$\hat{z}_1 = z_1 + z_4 + z_5,$$

$$\hat{z}_2 = z_2 + 2z_4,$$

$$\hat{z}_3 = z_3 + z_5,$$

illetve az ezekből adódó $\hat{t}_i = e^{\hat{z}_i}$ $(i=1, 2, 3)$ új változókat, ezekkel az eredetivel ekvivalens, s már csak $\hat{m}=3$ változós feladat (melyre $\hat{m}<n$ teljesül) írható fel:

$$\min(c_1 \hat{t}_1 \hat{t}_2^{1,5} \hat{t}_3^{-1} + c_2 \hat{t}_2^{-0,5} \hat{t}_3),$$

feltéve, hogy

$$\hat{\mathbf{t}} = (\hat{t}_1, \hat{t}_2, \hat{t}_3) > \mathbf{0},$$

$$c_3 \hat{t}_1^{0,5} \hat{t}_2 \hat{t}_3^{-2} + c_4 \hat{t}_1^{3,5} \hat{t}_2^{-2} \hat{t}_3^{2,5} \leqq 1.$$

Hasonlóan lehet redukálni a feladatot új változók bevezetésével, ha $n>m$, de $\mathbf{A}$ rangja m-nél kisebb.

2. *Megjegyzés.* Az egyensúlyi egyenleteknek a 3.1. lemma b) részében kimondott összefüggősége egyben összefüggést ad a (3.1) jobb oldalán szereplő, $\mathbf{y}$-tól függő kifejezések között is. Ez azt jelenti, hogy az optimális $\mathbf{y}$ megoldásoknak a normalitási és ortogonalitási feltételeken kívül további egyenleteket is ki kell elégíteniük (amelyeket már a megoldás megkezdése előtt felírhatunk), mégpedig annyit, ahány független lineáris kapcsolat van az egyensúlyi egyenletek együttható vektorai között. Ezt az észrevételt jól lehet hasznosítani a duálfeladat algoritmikus megoldásánál, például úgy, hogy az optimum közelében a kevésbé pontos y_i komponenseket (ezek általában a 0-hoz közeli komponensek) kifejezzük a többiekkel, aminek többszöri iteratív alkalmazása a konvergenciát az optimum közelében (ahol a 2. részben leírt gradiens vetítési algoritmus általában már lelassul) lényegesen meggyorsíthatja.

3. *Megjegyzés.* A 3.1. lemma a) része is lényeges a numerikus megoldás szempontjából. Ugyanis a feltétel, hogy minden m elemű oszloprendszer lineárisan független, a gyakorlatban általában (speciális ritka mátrixú feladatoktól eltekintve) teljesül, és a primál optimum ebből adódó egyértelműsége azt eredményezi, hogy a duál optimális megoldásból egyszerűen a (3.1) lineáris egyenletrendszer megoldásával adódik a primál optimális vektor (egyébként, ha a (3.1) rendszer határo-

zatlan, akkor megoldásai közül olyat kell keresni, amely kielégíti a primál feltételeket, tehát egy nem lineáris feltételrendszert és az egyensúlyi feltételeket kielégítő **z**-t kell keresni, ami számítástechnikailag lényegesen nehezebb az előbbi esetnél).

A $\log v(\mathbf{y})$ logaritmikus duálfüggvény $\mathbf{y}>\mathbf{0}$-ra, tehát a duál megengedett halmaz belsejében differenciálható, parciális deriváltjai:

$$\frac{\partial \log v(\mathbf{y})}{\partial y_i} = \begin{cases} c_i - \log y_i - 1, & \text{ha} \quad i \in J[0] \\ c_i - \log \dfrac{y_i}{\lambda_k(\mathbf{y})}, & \text{ha} \quad i \in J[k]. \end{cases} \tag{3.2}$$

Könnyen látható viszont, hogy $\log v(\mathbf{y})\ \exists_i : y_i = 0$ esetén, tehát a duál halmaz határán nem differenciálható (de itt is folytonos a (D) megfogalmazásában leírt kiterjesztést alkalmazva). Ez az egyik oka a duálfeladat numerikus megoldásánál fellépő nehézségeknek. Jelöljük D_0-lal az olyan duál megengedett pontok halmazát, melyekben a primál célfüggvénynek megfelelő komponensek pozitívak, s minden primál feltételnek megfelelő változó csoport vagy csupa pozitív, vagy csupa 0 komponensből áll. Már beláttuk, hogy optimális megoldásként csak a D_0 elemei jönnek szóba. Egy figyelemre méltó, de igen könnyen belátható tulajdonságot mond ki a következő lemma:

3.2. LEMMA. A D_0 határpontjaiban a D_0 belsejébe mutató irányok mentén létezik a $\log v(\mathbf{y})$ irány menti deriváltja.

Bizonyítás. Legyen $\mathbf{y}^0 D_0$ egy határpontja, s $\mathbf{d}$ egy az $\mathbf{y}^0$-ból a D_0 belsejébe mutató egységvektor (tehát $d_i > 0$ $y_i^0 = 0$ esetén). Az $\mathbf{y} = \mathbf{y}^0 + \alpha \mathbf{d}$, $\alpha > 0$ félegyenes mentén $\mathbf{y}^0$-hoz közeledve az $\mathbf{y}$ pont $\mathbf{y}^0$ 0 értékű komponenseinek megfelelő koordinátái:

$$y_i = \alpha d_i, \quad \text{ahol} \quad \alpha \to 0, \quad \text{amint} \quad \mathbf{y} \to \mathbf{y}^0.$$

Így, tekintetbe véve a $\log v(\mathbf{y})$ (1.1)-beli alakját az $\dfrac{1}{\alpha}[\log v(\mathbf{y}^0 + \alpha \mathbf{d}) - \log v(\mathbf{y}^0)]$ különbségi hányados egyrészt az $y_i^0 \neq 0$ komponenseknek megfelelő, $\alpha \to 0$ esetén véges határértékhez tartó tagokból, másrészt az $y_i^0 = 0$ koordinátákhoz tartozó

$$-\frac{1}{\alpha}\left[0 - \alpha d_i \left(c_i - \log \frac{d_i}{\sum\limits_{i \in J[k]} d_j}\right)\right] = d_i \left(c_i - \log \frac{d_i}{\sum\limits_{i \in J[k]} d_j}\right)$$

konstans tagokból tevődik össze, s így a határértéke $\alpha \to 0$ esetén valóban létezik.

Megjegyzés. A duál megengedett halmaz határpontjai közül csak a D_0-beliek jönnek számításba optimális pontként. A 3.2. lemma szerint viszont D_0 határpontjaiban létezik a D belsejébe mutató bármely irány mentén vett irány menti derivált.

Ez azt jelenti, hogy a megengedett tartomány belsejéből egy határon levő optimális pont felé haladva a haladási irány menti derivált stabilitást mutat, véges határérték felé tart. Ezért az algoritmusban a vetített gradiens menti előrehaladás során tetszőlegesen megközelíthető a tartomány határa, s ez fontos az eljárás működése szempontjából.

4. A módszer részletezése, lehetőségek a konvergencia gyorsítására, számítástechnikai tapasztalatok

Az algoritmus megindítása

Feltételezzük, hogy a megoldandó feladatban $n>m$ (ha az eredeti feladatban $n\leqq m$ lenne, akkor azt redukálni kell a 3.1. lemma utáni 1. megjegyzésnek megfelelően). Ha nem áll rendelkezésre a duál feladat egy megengedett megoldása, akkor egy első fázist kell végrehajtani, amely a lineáris feltételrendszer miatt megegyezik a szimplex módszer első fázisával. Ennek beépítése a programba nem jelent különösebb nehézséget.

Irány menti maximalizálás

A megengedett megoldásból elinduló, egymást követő iterációkban a (2.1) szerinti $\mathbf{d}^*=\mathbf{H}\nabla \log v(\mathbf{y})$ vetített gradiens irányban történik előrehaladás a $\log v(\mathbf{y})$ célfüggvény maximális növekedését biztosító lépéshosszal. A $\mathbf{d}^*$ irányvektor komponensei között általában pozitívak és negatívak is vannak, ezért először meg kell határozni azt a $(\theta_{\min}, \theta_{\max})$ intervallumot, melyre

(4.1) $$\mathbf{y}_1=\mathbf{y}_0+\theta\mathbf{d}^*\geqq\mathbf{0} \text{ teljesül.}$$

Gyakorlatilag nem engedhető meg $\mathbf{y}_1$ komponenseire a 0 elérése — ezt a problémát figyelmen kívül hagyja NIJKAMP [10]-beli tárgyalása —, mert a derivált a határon nem létezik. (4.1) helyett vehetnénk pl. a

(4.2) $$\mathbf{y}_1=\mathbf{y}_0+\theta\mathbf{d}^*\geqq\varepsilon_0\mathbf{1}$$

feltételt, ahol $\mathbf{1}=(1, \ldots, 1)'$ és ε_0 kis pozitív szám. Azonban célszerűbbnek látszik ehelyett az $\frac{y_1^i}{\lambda_k(\mathbf{y}_1)}$ hányadosokra tenni a megkötést, mivel a $\log v(\mathbf{y})$ függvényben és gradiensében ezeknek a logaritmusa szerepel. Tehát — a $\lambda_0(\mathbf{y})=1$ jelölés bevezetésével —:

(4.3a) $$\frac{y_1^i}{\lambda_k(\mathbf{y}_1)}\geqq\varepsilon_0, \quad i\in J[k],\ k=1, \ldots, p,$$

ahol

$$\mathbf{y}_1=\mathbf{y}_0+\theta\mathbf{d}^*.$$

(4.3a)-t átrendezve kapjuk:

(4.3b) $$\lambda_k(\mathbf{y}_0)\varepsilon_0-y_0^i\leqq\theta[d_i^*-\varepsilon_0\lambda_k(\mathbf{d}^*)].$$

Innen adódik:

$$\frac{\lambda_k(\mathbf{y}_0)}{\lambda_k(\mathbf{d}^*)}\frac{\varepsilon_0-\frac{y_0^i}{\lambda_k(\mathbf{y}_0)}}{\frac{d_i^*}{\lambda_k(\mathbf{d}^*)}-\varepsilon_0}\leqq\theta, \quad \text{ha} \quad \frac{d_i^*}{\lambda_k(\mathbf{d}^*)}>\varepsilon_0,$$

és a fordított egyenlőtlenség teljesül, ha $\frac{d_1^*}{\lambda_k(\mathbf{d}^*)}<\varepsilon_0$. (A gyakorlatban $d_i^*/\lambda_k(\mathbf{d}^*)=\varepsilon_0$

esetleges fellépése ε_0 kis módosításával mindig elkerülhető.) Így most már a θ-ra megengedhető intervallum határai:

$$\theta_{\min} = \max_{\substack{i:\, d_i^*/\lambda_k(\mathbf{d}^*)>\varepsilon_0 \\ i\in J(k) \\ k=1,\ldots,p}} \frac{\lambda_k(\mathbf{y}_0)}{\lambda_k(\mathbf{d}^*)} \cdot \frac{\varepsilon_0 - y_0^i/\lambda_k(\mathbf{y}_0)}{d_i^*/\lambda_k(\mathbf{d}^*) - \varepsilon_0}, \tag{4.4a}$$

$$\theta_{\max} = \min_{\substack{i:\, d_i^*/\lambda_k(\mathbf{d}^*)<\varepsilon_0 \\ i\in J[k] \\ k=1,\ldots,p}} \frac{\lambda_k(\mathbf{y}_0)}{\lambda_k(\mathbf{d}^*)} \cdot \frac{\varepsilon_0 - y_0^i/\lambda_k(\mathbf{y}_0)}{d_i^*/\lambda_k(\mathbf{d}^*) - \varepsilon_0}. \tag{4.4b}$$

Ha a $d_i^*/\lambda_k(\mathbf{d}^*)-\varepsilon_0$ komponensek mind egyforma előjelűek, akkor $\theta_{\min}$ és $\theta_{\max}$ közül az egyik definiálatlan, tehát az egyik irányban nincs korlátozás θ-ra. Ekkor ebben az irányban addig hajtunk végre lépéseket fokozatosan növekvő hosszúsággal, amíg a $\log v(\mathbf{y})$ csökkenővé válik, s ekkor az utolsó lépésnek megfelelő θ adja a korlátot ebben az irányban.

A kapott $[\theta_{\min}, \theta_{\max}]$ intervallumbeli maximalizálásra többféle eljárás használható. Mivel a $\log v(\mathbf{y})$ függvény konkáv, egy egyszerű felező eljárás, vagy egy „aranymetszési" kereső eljárás is célravezető. NIJKAMP egy harmadik módszert javasol, amelyben a $\log v(\mathbf{y})$ függvényt iterációnként a $\mathbf{d}^*$ irány mentén kvadratikus függvénnyel közelíti, ezt a $\theta_{\min}, \theta_{\max}$ és $\theta_{\text{int}} = \frac{1}{2}(\theta_{\min}+\theta_{\max})$ értékeknek megfelelő $\mathbf{y}_1=\mathbf{y}_0+\theta\mathbf{d}^*$ pontokban felvett függvényértékekre illesztve, s a kapott parabola maximumhelyével közelíti a keresett maximumot. Számítási tapasztalataim viszont azt mutatják, hogy egy egyszerű felező eljárás a leggyorsabb — amely mindig a felezőpontbeli irány menti derivált előjele alapján dönt a továbbhaladásról —, s ennek az lehet az oka, hogy a $\log v(\mathbf{y})$ függvény speciális alakja miatt függvényérték számítás esetén a gradiens kiszámítása már nagyon kevés pótlólagos időráfordítást igényel.

„Vetítés a határra"

Könnyen eldönthető egy irány menti maximalizálás kezdetén — egyszerűen a $\theta_{\min}, \theta_{\max}$ végpontokban felvett irány menti derivált érték alapján —, hogy a maximum az ε_0-lal megadott határon van-e. Ha a maximumot szolgáltató $\mathbf{y}^\circ$ határpont, azaz valamely i-re $y_i^0/\lambda_k(\mathbf{y}^0)=\varepsilon_0$ — egyszerre több y_j^0 komponens is a határra kerülhetne elvben, de numerikus számításnál ez nem fordul elő a kerekítési hibák miatt —, s az e pontban kiszámított $\mathbf{d}^*$ irányvektor a

$$y_i/\lambda_k(\mathbf{y}) \geqq \varepsilon_0, \quad i\in J[k],\ k=1,\ldots,p \tag{4.5}$$

feltételek által meghatározott tartományból kifelé mutat, akkor a $\log v(\mathbf{y}^0)$ gradienst a duál feltételeken kívül még az $y_i/\lambda_k(\mathbf{y})=\varepsilon_0$ feltételre is vetítjük (az $y_i/\lambda_k(\mathbf{y})$-t ε_0 szinten rögzítjük). Az új $\mathbf{H}'$ vetítési mátrix a régiből viszonylag egyszerűen kapható, a következő módon.

Legyen $\mathbf{s}_i=(1-\varepsilon_0)\mathbf{e}_i-\sum_{j\in J[k]}\varepsilon_0\mathbf{e}_j$, ahol $k: i\in J[k]$, és $\mathbf{e}_j$ a j-edik egységvektort jelöli. Ekkor az $y_i/\lambda_k(\mathbf{y})\geqq\varepsilon_0$ feltétel ekvivalens $\mathbf{s}_i'\mathbf{y}\geqq 0$-val. Ezért az, hogy egy az $\mathbf{s}_i'\mathbf{y}=0$-val jellemzett határpontban a $\mathbf{d}^*$ irányvektor a (4.5) feltételek által megha-

tározott tartományból kifelé mutat, ekvivalens azzal, hogy $\mathbf{s}_i'\mathbf{d}^*<0$. Továbbá most a duálfeladat $\mathbf{A}^*$ vetítési mátrixa az $\mathbf{s}_i'$ sorvektorral egészül ki, s így:

$$\mathbf{H}' = \mathbf{E}-(\mathbf{A}^{*\prime}, \mathbf{s}_i)\left[\begin{pmatrix}\mathbf{A}^*\\ \mathbf{s}_i'\end{pmatrix}(\mathbf{A}^{*\prime}, \mathbf{s}_i)\right]^{-1}\begin{pmatrix}\mathbf{A}^*\\ \mathbf{s}_i'\end{pmatrix} = \mathbf{E}-(\mathbf{A}^{*\prime}, \mathbf{s}_i)\mathbf{A}_2^{-1}\begin{pmatrix}\mathbf{A}^*\\ \mathbf{s}_i'\end{pmatrix},$$

ahol

$$\mathbf{A}_2 = \begin{bmatrix}\mathbf{A}^*\mathbf{A}^* & \mathbf{0}\\ \mathbf{0} & \mathbf{s}_i'\mathbf{s}_i\end{bmatrix} + \begin{bmatrix}0\\ \vdots\\ 0\\ 1\end{bmatrix}[\mathbf{s}_i'\,\mathbf{A}^{*\prime}, 0] + \begin{bmatrix}\mathbf{A}^*\mathbf{s}_i\\ \mathbf{0}\end{bmatrix}[0, \ldots, 0, 1].$$

Így az egy diáddal növelt mátrix inverzének ismert

$$(\mathbf{A}+\mathbf{u}\mathbf{v}^*)^{-1} = \mathbf{A}^{-1}-\frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^*\mathbf{A}^{-1}}{1-\mathbf{v}^*\mathbf{A}^{-1}\mathbf{u}}$$

képlete alapján $\mathbf{A}_2^{-1}$ két lépésben könnyen kiszámítható az eredeti vetítési mátrixban szereplő $(\mathbf{A}^*\mathbf{A}^{*\prime})^{-1}$-ből.

Tehát a módosított vetítési mátrix újabb invertálás nélkül kiszámítható a régiből. Ha a $\mathbf{H}'$-vel számított vetített gradiens menti $\mathbf{y}_1$ maximum pontban az eredeti $\mathbf{d}^*=\mathbf{H}\nabla\log v(\mathbf{y}_1)$ vetített gradiens $\mathbf{s}_i$ irányú vetülete, azaz $\mathbf{d}^*\mathbf{s}_i$ ismét <0, akkor az $y_i/\lambda_k(\mathbf{y})=\varepsilon_0$ rögzítést fenntartjuk, egyébként pedig feloldjuk (ha esetleg egy más komponens kerül a határra, s a neki megfelelő $\mathbf{d}^*\mathbf{s}_j$ komponens <0, akkor erre bevezetjük az előbbi rögzítést).

A tartósan nullához közeli komponensek kinullázása

A geometriai programozási feladat egyik jellegzetessége és a megoldásával kapcsolatos problémák egyik forrása az, hogy az optimális duálváltozók között lehetnek nulla értékűek — pl. ha a $\mathbf{t}^0$ optimális primál pontban egy primál feltétel nem aktív, akkor az ehhez tartozó optimális duál változók nulla értékűek —. Ezek megtalálását és a további számítás egyszerűsítését célozza egy „kinullázó" eljárás, amely egy komponenst 0-nak rögzít, ha az három egymás utáni iteráció alatt kis értékű ($\leqq\varepsilon_4$) marad, s a hozzá tartozó komponenscsoport összes tagja is elég kicsi ($\leqq 100\varepsilon_4$). A nulla értékű, rögzített y_i komponenseknek megfelelő $\mathbf{A}^*$-beli oszlopok ezután elhagyhatók, s a feladat egy kisebb méretűre redukálódik.

Elképzelhető, hogy egy ilyen 0-nak rögzítés téves. Ez akkor derül ki, amikor a kapott „optimális" duálmegoldásból a (3.1) lineáris egyenletrendszer — az ún. egyensúlyi egyenletek — segítségével az optimális primálmegoldást akarjuk kiszámítani. Legyen ugyanis N a 0-nak rögzített y_i komponenscsoportok indexhalmaza, s így az egyensúlyi egyenletek rendszere:

$$\sum_{j=1}^{m} a_{ij}z_j = \begin{cases}\log[y_i v(\mathbf{y})]-c_i, & i\in J[0],\\ \log[y_i/\lambda_k(\mathbf{y})]-c_i, & i\in J[k],\ k\notin N.\end{cases} \tag{4.6}$$

Az itt szereplő $\mathbf{y}$ vektor az eredeti duálfeladatból a kinullázott komponenseknek megfelelő változók és az együtthatómátrix ezekhez tartozó oszlopainak elhagyásával

kapott redukált feladat optimális megoldása, mégpedig pozitív optimális megoldása. Ez a redukált feladat, s a hozzá tartozó primál feladat a következő:

$$(4.7)\qquad \begin{cases} \textit{Primál feladat:} & \textit{Duál feladat:} \\ \min g_0(\mathbf{t}) & \max \log v(\mathbf{y}) = \\ g_k(\mathbf{t}) \leqq 1, & \sum\limits_{i \in J[0]} y_i(c_i - \log y_i) + \sum\limits_{\substack{k=1 \\ k \notin N}}^{p} \sum\limits_{i \in J[k]} \left[c_i + \log \frac{\lambda_k(\mathbf{y})}{y_i} \right], \\ k = 1, \ldots, p,\ k \notin N & \mathbf{B}^*\mathbf{y} = \mathbf{0}, \end{cases}$$

ahol $\mathbf{B}^*$ az $\mathbf{A}^*$-ból az N-be tartozó indexű oszlopok elhagyásával adódó mátrix.

Az $\mathbf{y}$-ból a (4.6) segítségével kapott $\mathbf{z}$, ill. az ebből a $t_i = e^{z_i}$ helyettesítéssel kapott $\mathbf{t}$ vektor a (4.7)-ben szereplő primál feladat feltételeit biztosan kielégíti, de arra már nincs biztosíték, hogy a redukálás előtti primálfeladat $k \in N$-nek megfelelő pótlólagos feltételeit is kielégítse. Ha egy pótlólagos feltétel nem lesz kielégítve, tehát, ha $\exists k \in N: g_k(\mathbf{t}) > 1$, akkor a leírt „0-nak rögzítés" téves volt.

Ekkor vissza kell térni az algoritmusnak arra a pontjára, ahol a legutolsó kinullázás történt, ezt fel kell oldani, s innen folytatni az eljárást kisebb ε_4-gyel. Ilyen probléma ismételt felmerülése esetén az előzőhöz hasonlóan újabb nulla rögzítést (vagy rögzítéseket) kell feloldani. Számítástechnikai tapasztalataim szerint azonban ε_4 elég kis értéke ($10^{-8} - 10^{-12}$) esetén ilyen téves kinullázási veszély nincs.

Kérdés még az ε_0 megválasztása. ε_0 értékét $10^{-8} - 10^{-10}$ körülinek választhatjuk, de ε_0-t az algoritmus során szükség esetén csökkenthetjük. Ilyen csökkentésre akkor van szükség, ha egy y_i már többször elérte az $y_i/\lambda_k(\mathbf{y}) = \varepsilon_0$ által megadott „határt", s erre vetíteni kellett, de az $y_i < \varepsilon_4$ feltétel még nem teljesül (tehát y_i-t „kinullázni még nem lehet"). Azonban tapasztalataim szerint pl. $\varepsilon_0 = 10^{-10}$, $\varepsilon_4 = 10^{-10}$ esetén ε_0 ilyen csökkentésére nem volt szükség.

Az optimális primál megoldás számítása

Ha kielégítő pontossággal elértük a duál optimumot — megállási kritériumként kis értékű relatív függvény- és gradiensváltozás, továbbá kis gradiensnorma elérését alkalmazzuk —, akkor a következő feladat: az egyensúlyi feltételek felhasználásával egy optimális primál megoldást állítani elő. Tudjuk, hogy minden optimális $\mathbf{t}$ primál vektor, ill. az ebből $e^{z_i} = t_i$, $i = 1, \ldots, m$ helyettesítéssel kapott $\mathbf{z}$ vektor kielégíti a (3.1) egyensúlyi feltételrendszert. Gyakorlati feladatoknál mindig feltehető, hogy van primál optimális megoldás, s általában az is teljesül, hogy az $\mathbf{A}$ bármely m különböző oszlopa lineárisan független, s így a 3.1. lemma értelmében a primál feladatnak csak egy optimális megoldása van. Ez lényegesen megkönnyíti a helyzetet, mert így elég a (3.1) lineáris egyenletrendszer egyetlen $\mathbf{z}^0$ megoldását előállítani, míg egyébként a (3.1) feltételeket kielégítő megoldások közül kellene egy olyat kiválasztani, ami a nemlineáris primál feltételeknek is eleget tesz (ennek számítástechnikai megvalósítása sokkal nehezebb lenne). Mivel a (3.1) egyenletrendszer gyakorlatilag mindig túlhatározott — ezt fejezi ki az említett Lemma b) része —, s a duáloptimum csupán közelítő meghatározása miatt így egyenletei ellentmondóak, azért egy közelítő megoldása adható csak, amit a legkisebb négyzetek módszerével — az egyenletek hibái négyzetösszegét minimalizálva — határozunk meg.

Különböző módszerek a konvergencia gyorsítására

a) *A megoldás korrekciója.* A számítási hibák miatt az algoritmus folyamán előállított megoldások a lineáris duálfeltételeket csak bizonyos hibával elégítik ki. Ez a hiba halmozódik, s ezért időnként célszerű megvizsgálni a feltételek kielégítettségi szintjét, s szükség esetén korrekciót hajtani végre. A korrekció egyik kézenfekvő módja a legkisebb négyzetek módszere, amelynél a korrigálandó megoldás komponenseitől való eltérések négyzetösszegét minimalizáljuk a lineáris duálfeltételek kielégítése mellett. A korrekcióra elsősorban nagyobb méretű feladatok esetében van szükség, melyeknél a hiba gyorsabban halmozódik.

b) *Az egyensúlyi feltételek összefüggőségének hasznosítása.* A 3.1. lemma szerint a (3.1) egyensúlyi (lineáris) egyenletek együtthatóvektorai mindig lineárisan összefüggők. Már említettük a 3.1. lemma utáni 2. megjegyzésben, hogy ezt fel lehet használni a duál algoritmus konvergenciájának meggyorsítására. Ugyanis optimális $\mathbf{y}_0$ esetén a (3.1) jobb oldalán szereplő, **y**-tól függő kifejezések között a bal oldali együtthatóvektorok közötti lineáris függésnek megfelelő kapcsolat áll fenn.

Az ilyen független lineáris kapcsolatok számának megfelelő számú y_i komponenst fejezhetünk ki a többivel. Célszerű a legkisebb, tehát relatíve legpontatlanabb komponenseket fejezni ki. Feltehető (a 3.1. lemma utáni 1. megjegyzés értelmében), hogy az $m \times n$-es **A** exponens mátrix rangja m, s így $n-m$ y_i komponenst lehet a többivel kifejezni. Ennek megvalósítására, ha már elég közel vagyunk az optimumhoz, kiválasztjuk a lehető legkisebb y_i-kből álló olyan $n-m$ tagú együttest, hogy az **A** mátrixnak a komplementer indexhalmazhoz tartozó $m \times m$-es része, $\mathbf{A}_1$ reguláris legyen. Az $\mathbf{A}=(\mathbf{A}_1, \mathbf{A}_2)$ particionálás esetén tehát az $\mathbf{A}_2$-beli oszlopok kifejezhetők $\mathbf{A}_1$-beliek lineáris kombinációjaként:

$$\mathbf{a}_2^i = \mathbf{A}_1 \mathbf{g}^i, \quad \text{ahol} \quad \mathbf{g}^i = \mathbf{A}_1^{-1} \mathbf{a}_2^i.$$

Így az egyensúlyi feltételrendszer (3.1)-beli alakjában szereplő i-edik egyenlet jobb oldali kifejezését d_i-vel, az $\mathbf{A}_1$-nek megfelelő ilyen kifejezések vektorát **d**-vel jelölve az $\mathbf{A}_2$-nek megfelelő i-kre $d_i=(\mathbf{A}_1^{-1}\,\mathbf{a}_2^i)'\mathbf{d}$, és így ilyen i indexekre

$$\frac{y_i}{\lambda_k(\mathbf{y})} = e^{c_i} e^{d_i}. \tag{4.8}$$

Itt y_i új, korrigált értékének számításához a régi $\lambda_k(\mathbf{y})$ összeget használjuk fel.

Tapasztalataim szerint ezzel az eljárással bizonyos esetekben — amikor az optimális $\mathbf{y}^0$ egyes komponensei igen kis értékűek — jelentősen meg lehet gyorsítani a konvergenciát.

c) *A primál feltételek aktivizálása.* NIJKAMP [10]-ben javasolja a következő értelmű primál-duál lépések beiktatását. Ha a duál-iterációk során a célfüggvényérték már csak keveset változik, akkor a (3.1) alapján meghatározzuk a primál optimum egy $\hat{\mathbf{z}}$ közelítését. Feltételezzük, hogy már sikerült azonosítani az optimumban 0 értékű duálváltozókat, s így ezeket „kinulláztuk". A többi duálváltozóhoz tartozó primál feltételek az optimumban szükségképpen aktívak kell legyenek, ezeket a $\hat{\mathbf{z}}$ módosításával mesterségesen „aktivizáljuk". Ez úgy történik, hogy az aktivizálandó primál feltételi függvényeket $\hat{\mathbf{z}}$ körül sorbafejtve csak a lineáris tagokat vesszük figyelembe, ony egy lineáris egyenletrendszerből számíthatjuk a $\Delta\mathbf{z}$ korrekciós

vektor komponenseit. Ezután a $\mathbf{z}^1=\hat{\mathbf{z}}+\Delta\mathbf{z}$ felhasználásával a (3.1) alapján egy új duálmegoldást állítunk elő, s ismét duál iterációkat hajtunk végre.

Ezen elgondolással kapcsolatban több nehézség is felmerül. Az aktivizálásnál egyrészt nem garantált, hogy az eddig kielégített nem aktív korlátokat nem sértjük meg, másrészt nem biztos, hogy a célfüggvény javulni fog. Továbbá az aktivizált $\mathbf{z}^1$-ből kapott új duál vektor nem biztos, hogy megengedett, esetleg még korrigálni kell. Számítástechnikai tapasztalataim szerint ilyen értelmű primál-duál lépéseket nem érdemes beépíteni az algoritmusba. Viszont az optimumtól való távolság megbecslésére mindenképpen alkalmas a mindenkori legjobb duálmegoldáson vett duálfüggvényérték, s a hozzá a (3.1) segítségével számított primál-ponton adódó primál függvényérték közötti eltérés.

d) *Az irány menti maximalizálás módosítása.* Az algoritmus folyamán a **H** vetítési mátrix lényegében állandó, ezért a vetített gradiens számítása könnyű, a felhasznált idő túlnyomó részét az irány menti maximalizálásra — az ehhez szükséges függvénykiértékelésekre — kell fordítani. Így ennek a jobb, rugalmasabb megszervezésétől várható az algoritmus gyorsítása. A kezdő iterációknál a teljes $(\theta_{\min}, \theta_{\max})$ intervallumból kiindulva kell egyre jobban lokalizálni a maximum helyét az adott irány mentén; azonban az optimumhoz való közeledéssel — tapasztalataim szerint — bizonyos mértékig stabilizálódik az irány menti maximumot biztosító lépéshossz, s így elég az ennek megfelelő $\mathbf{y}_0$ pontot tartalmazó, a $(\theta_{\min}, \theta_{\max})$-nál kisebb intervallumra korlátozni a vizsgálatot.

A módszer számítógépes megvalósítása

Az előző részben leírt algoritmus számítógépes megvalósítására készített kb. 2000 utasításos GEOPR nevű program a főprogramon kívül 8 szubrutint tartalmaz, s ehhez csatlakoznak még 5 tesztfeladat adatait generáló INP1—INP5 szubrutinok. A főprogram hívja be a megoldandó feladat paramétereit és induló megoldását beállító szegmenst, majd a feladatot megoldó GRPROJ szubrutint is, a futást vezérlő különböző paraméterek megadása után (pl. hogy hogyan változtassuk az ε_0 küszöböt, milyen pontosan maximalizáljunk az egyes iterációkban, hány iterációnként írassuk ki a részeredményeket stb). A GRPROJ hívja a STEP szubrutint, amely egy duáliterációt — a megfelelő irány meghatározása, majd e mentén való maximalizálás — hajt végre. Az irány menti maximalizálást az LMAX1 szubrutin (ezt hívja a STEP), a $\log v(\mathbf{y})$ függvényérték és gradiense számítását a LOGF szubrutin végzi. A CORR és a NULL szubrutinok hajtják végre az előző részben tárgyalt korrekciót és kinullázást, míg a DEPEND az említett, lineáris összefüggőségre támaszkodó konvergencia-gyorsítást. A PRIML1 szubrutin a (3.1) egyensúlyi feltételek alapján közelítő optimális primál megoldást számít.

A programot 5 irodalmi teszt-feladaton ([2], [3], [4], [5, 396—401] és [9]) és egy kísérleti célú népgazdaságtervezési modell (leírása megtalálható [6] és [7]-ben) számítógépes realizációján futtattam. Ez az utóbbi feladat volt a legnagyobb méretű, itt primál alakban 30 változó, 35 feltétel, 90 additív kifejezés szerepelt. A munkával kapcsolatos számítógépes tapasztalatok, vagy az alkalmazási eredmények iránt jobban érdeklődő olvasó részletesebb leírást találhat [6]-ban.

IRODALOM

[1] AVRIEL, M., "On a decomposition for a special class of geometric programming problems", *Journal of Optimization Theory and Applications* **3** (1969) 392—409.
[2] AVRIEL, M., and WILDE, D. J., "Optimal condenser design by geometric programming", *Industrial and Engineering Chemistry, Process Design and Development* **6** (1967) 256—265.
[3] DANTZIG, G. B., JOHNSON, S. and WHITE, W., "A linear programming approach to the chemical equilibrium problem", *Management Science* **5** (1958) 38—43.
[4] DUFFIN, R., PETERSON, E. and ZENER, C., *Geometric Programming — Theory and Application* (John Wiley Publishing Company, New York—London—Sydney, 1967).
[5] HIMMELBLAU, D., *Applied Nonlinear Programming* (Mc Graw Hill Publishing Company, New York, 1972).
[6] KÁDAS, S., „A geometriai programozás egy megoldási módszere és néhány közgazdasági alkalmazása", doktori értekezés. Marx Károly Közgazdaságtudományi Egyetem, Budapest, 1974.
[7] KÁDAS, S., „A geometriai programozás és két gazdasági alkalmazása", *Szigma* **7** (1974) 17—24.
[8] KLAFSZKY, E., „Geometriai programozás és néhány alkalmazása", kandidátusi értekezés. Magyar Tudományos Akadémia, Budapest, 1973.
[9] MINE, H. and OHNO, K., "Decomposition of mathematical programming problems by dynamic programming and its application to block-diagonal geometric programs", *Journal of Mathematical Analysis and Applications* **32** (1970) 370—385.
[10] NIJKAMP, P., *Planning of industrial Complexes by Means of Geometric Programming* (Rotterdam Universitaire Pers, Rotterdam, 1972).

(Beérkezett: 1975. április 17.)
(Újra beérkezett: 1975. november 1.)

DR. KÁDAS SÁNDOR
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1502 BUDAPEST XI., KENDE U. 13–17.

A SOLUTION METHOD OF THE GEOMETRIC PROGRAMMING PROBLEM

S. KÁDAS

The dual problem of geometric programming consists of maximizing a concave function under linear equality constraints. It is a natural approach to apply the gradient projection method for the solution of this problem, but an adaptation of the method to special requirements encountered here is necessary. The paper presents an algorithm based on the gradient projection method and containing heuristic elements as well, and also the related computational experiences. The first part of the paper contains the formulation of the geometric programming problem and its duality theorem, the second part presents the scheme of the algorithm, the third part outlines the proofs of two simple but interesting and from the viewpoint of the algorithm useful properties of the geometric programming problem, while the last part contains the detailed description of the algorithm and presentation of computational experiences obtained.

Alkalmazott Matematikai Lapok **2** *(1976) 83—91*

GYÁRTÁSÜTEMEZÉS A GÉPBEÁLLÍTÁSI IDŐK MINIMALIZÁLÁSÁRA, ALTERNATÍV GÉPVÁLASZTÁS MELLETT

HEPPES ALADÁR és LUGOSI GÁBOR
Budapest

A cikk egy gyártásütemezési problémára közöl egzakt algoritmust. A munkákat két gépen úgy kell ütemezni, hogy a gépbeállítási idők összege minimális legyen. A munkák egyik halmaza kizárólag egy-egy gépen gyártható, a munkák másik halmaza mindkét, technológiailag különböző adottságú gépen megmunkálható.

Az ütemezési problémának egy gráfelméleti probléma feleltethető meg. Az optimális ütemezés meghatározása a gráfok egy halmazában minimális súlyú irányított *Hamilton-kör* keresésére vezethető vissza. Az adott tulajdonságú *Hamilton-kör* a szétválasztás és korlátozás módszerével állítható elő. Ennek algoritmusát egy numerikus példán keresztül szemléltetjük.

1. Bevezetés

Cikkünk témája egy nagy, kohászati gyárban a profilhajlító gépsorok gyártásütemezési rendszerének egy részproblémája. Egy üzemben két, különböző technológiai adottságú gép van. A munkák (termékek) egyik része csak az első gépen, a második része kizárólag a második gépen, míg a munkák harmadik csoportja mindkét gépen gyártható. A gépeken a megmunkálási idők az ütemezéstől függetlenek, viszont a gépbeállítási idők — gépenként — attól függnek, hogy a gyártásban melyik munkáról melyikre kell átállni. Az említett részproblémában a cél az volt, hogy a harmadik csoport munkáit úgy osszuk szét a két gépre, hogy a két gépen a gépbeállítási idők összege minimális legyen. Nyilvánvaló, hogy ez a probléma csak úgy oldható meg, ha az összes munka gyártási sorrendjét is meghatározzuk a két gépen.

Érdekes módon az ütemezéselmélet bő irodalma alig foglalkozik az alternatív gépmegválasztás problémájával, ha a gépek technológiai adottságai eltérőek, jóllehet ez a gyakorlatban nem ritka helyzet. Az irodalomból mindössze egy ilyen cikket ismerünk [2], ez ütemezéstől független gépbeállítási idők mellett a gépek egyenletes terhelésére közöl egy heurisztikus algoritmust.

2. A probléma

Legyen J az ütemezendő munkák halmaza, mely n számú munkát tartalmaz. A munkákat i egészekkel azonosítjuk ($1 \leqq i \leqq n$), így J egy indexhalmaz. A J halmaz három, páronként diszjunkt részhalmaz egyesítése:

$$J = J_A \cup J_B \cup J_{AB}, \quad \text{és}$$

$$J_A \cap J_B = J_A \cap J_{AB} = J_B \cap J_{AB} = \emptyset, \tag{2.1}$$

ahol J_A elemei a csak az A gépen gyártható, a J_B elemei a kizárólag a B gépen gyártható, és J_{AB} elemei a mindkét gépen megmunkálható munkák.

Jelöljük $s_{ij}^{(A)}$-val és $s_{ij}^{(B)}$-vel az i munkáról a j munkára átállásnál a gépbeállítási időt az A, illetve a B gépen. Az $s_{ij}^{(A)}$ csak akkor értelmezett, ha $i, j \in J_A \cup J_{AB}$ és $i \neq j$. Hasonlóan, $s_{ij}^{(B)}$-t csak $i, j \in J_B \cup J_{AB}$ és $i \neq j$-re értelmezzük. Így, ha $i, j \in J_{AB}$, két beállítási idő létezik, melyek általában nem egyeznek meg.

Egy *ütemezést* a következőképpen definiálhatunk a két gépre. Jelöljük S'_A és S'_B-vel az A és B gépre sorolt munkák halmazát. Azt, hogy egy ütemezésben a két gépen minden munkát egyszer, és csakis egyszer le kell gyártani, továbbá, hogy az A gépen a J_B elemei és a B gépen a J_A elemei nem munkálhatók meg, a

$$|S'_A| + |S'_B| = n, \quad \text{és} \quad S'_A \cap S'_B = \emptyset,$$

valamint a

$$J_B \cap S'_A = \emptyset \quad \text{és} \quad J_A \cap S'_B = \emptyset$$

relációkkal fejezhetjük ki. Egy ütemezésnek nevezzük az $S = (S_A, S_B)$ rendezett halmazpárt, ahol az S_A és az S_B halmazok az S'_A és az S'_B halmazok elemeinek egy permutációját képezik.

Legyen F az összes ütemezés halmaza. Minden $S \in F$ ütemezéshez egy $U(S)$ valós számot rendelünk hozzá, a *gépbeállítási idők összegét:*

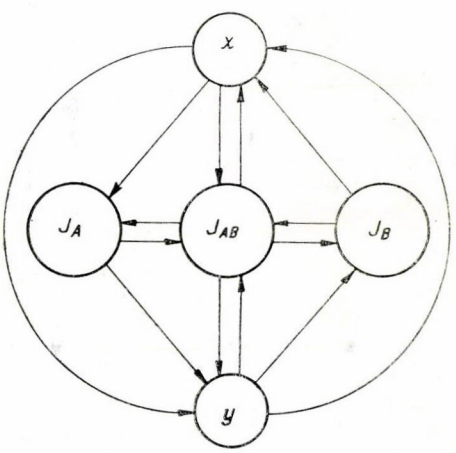
$$U(S) = \sum_{i=1}^{|S_A|-1} s^{(A)}_{\varrho(i)\varrho(i+1)} + \sum_{j=1}^{|S_B|-1} s^{(B)}_{\pi(j)\pi(j+1)}, \tag{2.2}$$

ahol $\varrho(i)$ az S_A rendezett halmaz i-edik eleme $(1 \leqq i \leqq |S_A|)$ és $\pi(j)$ az S_B rendezett halmaz j-edik eleme. $(1 \leqq j \leqq |S_B|)$.

Optimális ütemezésnek a

$$\min_{S \in F} U(S)$$

feltételnek megfelelő ütemezést nevezzük, azaz azt az ütemezést, amely az F halmazban a minimális $U(S)$ értéket veszi fel. Célunk egy olyan algoritmus konstruálása, mely egy optimális ütemezést előállít.



1. ábra

Ezt a kombinatorikus problémát átfogalmazzuk gráfeleméleti problémává a jobb áttekintés kedvéért. Tekintsünk egy $G(V, E)$ irányított gráfot (1. ábra), amelynek csúcsait a munkák J halmaza és két további pont képezik:

$$V = J \cup \{x, y\},$$

élei pedig:

$$\begin{aligned} E = {} & \big((u, v) | u, v \in J_A \cup J_{AB}\big) \cup \\ & \big((u, v) | u, v \in J_B \cup J_{AB}\big) \cup \\ & \big((u, v) | u, v \in J_{AB} \cup \{x, y\}\big) \cup \\ & \big((x, u), (u, y) | u \in J_A\big) \cup \\ & \big((u, x), (y, u) | u \in J_B\big) \cup \\ & \{(x, y), (y, x)\}. \end{aligned}$$

Könnyen belátható, hogy minden megvalósítható ütemezésnek a G gráfban egy olyan *Hamilton-kör* felel meg, amely x-ből y-ba vezető íve során nem érint J_B-beli pontot és y-ból x-be vezető ívén nincs J_A-beli csúcs, s ez a megfeleltetés kölcsönös. Jelöljük az ilyen *Hamilton-kör*ök halmazát H-val.

Ha egy H-beli *Hamilton-kör* éleihez x-től y-ig a megfelelő $s_{ij}^{(A)}$, y-tól x-ig a megfelelő $s_{ij}^{(B)}$ súlyokat rendeljük, és ezeket összegezzük, a teljes átállítási időt kapjuk.

Célszerűnek látszik a gráf éleihez ezeket a súlyokat eleve hozzárendelni, hogy a *Hamilton-kör*öket könnyen értékelhessük, problémát jelent azonban, hogy a J_{AB} halmaz pontjaira illeszkedő élek esetében nem rögzíthetjük a súlyt, hiszen az az illető élnek a *Hamilton-kör*ben betöltött szerepétől függ, nevezetesen attól, hogy a megfelelő munka az A vagy B gépre sorolódik.

A gráf éleinél ezért speciális súlyozást alkalmazunk: az x és y pontokra illeszkedő élek súlya 0, a többi élhez pedig egy vagy két értéket rendelünk: $s_{ij}^{(A)}$ súlyt, ha $(i,j)\in S'_A$, $s_{ij}^{(B)}$ súlyt, ha $(i,j)\in S'_B$ és mindkettőt, ha mindkét feltétel teljesül.

Célunk a minimális súlyú H-beli (azaz ütemezésként megvalósítható) *Hamilton-kör* meghatározása, ahol a kör súlyát az élek megfelelően választott súlyainak összege adja. Elvben járható út volna a J_{AB}-hez tartozó pontokat minden lehetséges ($2^{|J_{AB}|}$) módon eleve az S'_A illetve a S'_B halmazokba sorolni — ez a súlyok kétértékűségét is megszünteti — majd a minimális súlyú *Hamilton-kör* meghatározását egy-egy $|J|+2$ méretű utazó ügynök problémaként fogni fel, ezt az utat azonban a tetemes számítási igény miatt el kell vetni.

Az alább bemutatott algoritmus számítási igénye jóval kisebb, hozzávetőlegesen egyetlen $|J|+2$ méretű utazó ügynök probléma megoldásának számítási igényével egyenértékű.

3. Az algoritmus

A minimális súlyú H-beli *Hamilton-kör*t szétválasztás és korlátozás módszerével állítjuk elő, hasonló módon, mint ahogyan LITTLE, MURTY, SWEENEY és KAREL az utazó ügynök probléma megoldását meghatározták [1].

Az algoritmus a $H_{0,0}=H$ halmazból és a $G_{0,0}=G$ gráfból indul ki. A szétválasztásra kerülő halmazok H részhalmazai, de minden ilyen részhalmazhoz készítünk egy gráfot is, amely a részhalmazhoz tartozó *Hamilton-kör*ökön kívül lehetőleg minél kevesebb nem H-beli *Hamilton-kör*t tartalmaz. Az élrögzítéseken és élkizárásokon alapuló algoritmus fokozatosan kizárja a nem H-beli *Hamilton-kör*öket, egyértelművé teszi az élek súlyozását, és pontossá az egy-egy részhalmazhoz tartozó *Hamilton-kör*ök súlyára adott alsó becslést.

A részhalmazokat ismételten szétválasztjuk két diszjunkt részhalmazra oly módon, hogy az egyik részhalmaz egy e élt tartalmazó irányított *Hamilton-kör*ökből áll, és a másik részhalmaz az e élt nem tartalmazó irányított *Hamilton-kör*ökből áll. Az algoritmus egy általános lépésénél legyen adva egy

$$H_{ij}(e_1, \ldots, e_i, \bar{f}_1, \ldots, \bar{f}_j), \quad i \leqq |J|$$

halmaz, mely azon irányított *Hamilton-kör*ökből áll, amelyek

α) az $e_1, \ldots, e_i \in E$ éleket tartalmazzák,

β) az $f_1, \ldots, f_j \in E$ éleket nem tartalmazzák,

γ) elemei a H halmaznak, azaz megvalósítható ütemezésnek felelnek meg.

Célunk, hogy a H_{ij}-hez tartozó *Hamilton-kör*ök súlyára minél jobb alsó becslést kapjunk, ezért olyan $G_{ij}(V, E_{ij})$ gráfot konstruálunk, amely az összes H_{ij}-beli *Hamilton-kör*t tartalmazza, és lehetőleg minél kevesebb továbbit. A konstrukciós eljárás abból áll, hogy G-ből három típusú kizárás sorozattal minél több élt kizárunk.

a) Az α) és β) feltételeknek ellentmondó élek kizárása:

$$(u, v) \notin E_{ij}, \quad \text{ha} \quad (u, v) \in \{f_1, \ldots, f_j\} \tag{3.1}$$

és

$$(u, v) \notin E_{ij}, \quad \text{ha} \quad (u, s) \in \{e_1, \ldots, e_i\}, \quad \text{de} \quad s \neq v, \tag{3.2}$$

$$\text{vagy} \quad (t, v) \in \{e_1, \ldots, e_i\}, \quad \text{de} \quad t \neq u.$$

b) A gráf további egyszerűsítésére a „rövid záródást" okozó élek kizárása:

Nem lehet az (u, v) él E_{ij}-ben, ha $i < |V| - 1$, és az élhalmaz egy irányított kört tartalmaz.

E feltétel vizsgálatának megkönnyítésére vezessünk be egy $|V|$ méretű $D_i \cdot (e_1, \ldots, e_i)$ kvadratikus mátrixot, amelyben $d_{uv} = 1$, ha $G'(V, \{e_1, \ldots, e_i\})$-ben az u csúcsból a v csúcsba irányított út vezet, és $d_{uv} = 0$ egyébként.

A kizárási feltétel most így fogalmazható:

$$(u, v) \notin E_{ij}, \quad \text{ha} \quad d_{vu} = 1. \tag{3.3}$$

c) Olyan élek kizárása, amelyek az α) alatti rögzítések következtében H-beli *Hamilton-kör*ben nem fordulhatnak elő. (Ilyen kizárásra akkor adódik lehetőség, ha a legutóbbi él-rögzítés „még el-nem-kötelezett" munkát pl. az A gépre sorol. Ekkor ugyanis kizárható minden olyan él, amely a megfelelő pontból S'_B-beli felé vagy felől halad, továbbá egyértelművé válnak a pontra illeszkedő élek súlyai.)

A kizárás szabálya a következő:

$t \in J_A$, $u \in J_{AB}$, $v \in J_B$ esetén,

ha $d_{uv} = 1$, vagy $d_{vu} = 1$, akkor $(t, u) \notin E_{ij}$, $(u, t) \notin E_{ij}$, továbbá $s_{uz}^{(A)}$ és $s_{zu}^{(A)}$ törlendő minden $z \in J_{AB}$-re,

ha pedig $d_{tu} = 1$, vagy $d_{ut} = 1$, akkor $(u, v) \notin E_{ij}$, $(v, u) \notin E_{ij}$, továbbá $s_{uz}^{(B)}$ és $s_{zu}^{(B)}$ törlendő minden $z \in J_{AB}$-re.

Bármely $C \in H_{ij}$ *Hamilton-kör* súlyára (így a minimális súlyú H-beli-re is) kaphatunk egy $A(H_{ij})$ alsó becslést, ha a kapott $G_{ij}(V, E_{ij})$ gráf minden csúcsára megkeressük a minimális súlyú kifutó élet — ha egy élhez két érték is tartozik, mindkettőt figyelembe véve — és ezen élek súlyát (mindkét súlyát) a minimális súllyal redukáljuk. Ezután az így redukált gráfon — a fentihez hasonlóan — minden csúcsra megkeressük a minimális súlyú befutó élt, és ezen élek súlyát ezzel a minimális súllyal redukáljuk.

A redukációk összege egy alsó becslést szolgáltat, mivel minden *Hamilton-kör* minden V-beli csúcsnál egy befutó és egy kifutó élt kell tartalmazzon, és a kör értéke a redukált élen sem lehet negatív.

A fenti alsó becslés csak akkor nem értelmezhető, ha van olyan csúcs, amelyből nem fut ki, vagy amelybe nem fut be él. Ekkor azonban ebben a gráfban már nem is alakítható ki *Hamilton-kör*, így $H_{ij} = \emptyset$, tehát az algoritmusnak ez az ága lezárul.

Könnyen belátható, hogy az $A(H_{ij})$ becslés ismeretében a redukált súlyú G_{ij} gráfból a

$$H_{i+1,j}(e_1, \ldots, e_i, e, f_1, \ldots, f_j)$$

alsó becslése közvetlenül is meghatározható. Töröljük a redukált súlyú G_{ij} gráfból azokat az éleket, amelyek nem elemei a $G_{i+1,j}$ gráfnak. Az így konstruált gráfon hajtsuk végre a redukciós eljárást. Ha a (további) redukciók összegét $R_{i+1,j}$-vel jelöljük, akkor az

$$A(H_{i+1,j}) = A(H_{i,j}) + R_{i+1,j}$$

egyenlőség teljesül. Hasonlóan

$$A(H_{i,j+1}) = A(H_{i,j}) + R_{i,j+1}.$$

A bevezetett jelölések mellett a minimális súlyú *Hamilton-kör*t előállító algoritmust az alábbi lépésekben foglalhatjuk össze:

1. Lépés. Kezdőértékadás: $i=0, j=0$, $H_{0,0}=H$, $G_{0,0}=G$ $A(H_{0,0})=0$, $\mathrm{OPT}=\infty$ és minden $d_{ij}=0$ legyen. A bevont élek száma i, a kizártaké j, OPT pedig az eddig talált legjobb megoldás értéke. A megoldást magát az ε vektorban tároljuk.

Az $A(H_{0,0})$ alsó becslés 0, mert G minden csúcsára az x vagy y csúcsból pozitívan is, negatívan is illeszkedik zéró súlyú él.

Helyezzük el G súlyait egy W csúcs-mátrixban, amelynek egy W_{st} eleme az (s, t) él súlya legyen, vagy ∞, ha nincs G-ben megfelelő él. A J_{AB} pontjainak megfelelő sorokban és oszlopokban elemenként két értéket tárolunk: $s_{ij}^{(A)}$-t és $s_{ij}^{(B)}$-t.

Szétválasztás után mindig azzal a részhalmazzal foglalkozunk először, amelyik él-rögzítéssel keletkezett. Így bármely szétválasztási szinten legfeljebb egy „félretett" részhalmaz fordulhat elő. A szétválasztás szintjét $i+j$, a vizsgálat alatt álló részhalmaz származását pedig az $\alpha(k)$ számok adják meg. Utóbbi értelmezése:

$$\alpha(k) = \begin{cases} 0, \text{ ha a } k \text{ szinten az él-rögzítés van érvényben,} \\ 1, \text{ ha a } k \text{ szinten az él-kizárás van érvényben.} \end{cases}$$

2. Lépés. Keressük meg azt az $e \in G_{ij}$ élt, amelyre

$$H_{i,j+1}(e_1, \ldots, e_i, \bar{f}_1, \ldots, \bar{f}_j, \bar{e}) = \emptyset,$$

vagy ha ilyen e él nem létezik, akkor azt, amelyre e a

$$\max_{e \in G_{ij}} A(H_{i,j+1})$$

alsóbecslést biztosítja. Számítástechnikailag állítsuk elő az

$$\omega = \{(u_1, v_1), \ldots, (u_s, v_s) | W_{u_t v_t} = 0,\ 1 \leqq t \leqq s\}$$

indexpárhalmazt. Határozzuk meg azt az r indexet $(1 \leqq r \leqq s)$, amely a

$$\theta = \max_{(u_t, v_t) \in \omega} \left(\min_{u_h \neq u_t} W_{u_h v_h} + \min_{v_h \neq v_t} W_{u_h v_h} \right)$$

értéket biztosítja. Legyen $e=(u_r, v_r)$.

3. Lépés. Legyen $i := i+1$, $e_i := e$ és $\bar{f}_{j+1} := \bar{e}$. Válasszuk szét e alapján a $H_{i-1,j}$ halmazt a

$$H_{i,j}(e_1, \ldots, e_{i-1}, e_i = e, \bar{f}_1, \ldots, \bar{f}_j)$$

és

$$H_{i-1,j+1}(e_1, \ldots, e_{i-1}, \bar{f}_1, \ldots, \bar{f}_j, \bar{f}_{j+1} = \bar{e})$$

részhalmazokra. Ekkor

$$A(H_{i-1,j+1}) = \begin{cases} A(H_{i-1,j})+\theta, & \text{ha} \quad \theta < \infty, \\ \infty \quad \text{egyébként} \end{cases}$$

lesz. Ha $A(H_{i-1,j+1})=\infty$, értelemszerűen $H_{i-1,j+1}=\emptyset$. A $G'(V, \{e_1, \ldots, e_i\})$ gráfból határozzuk meg a D mátrixot.

Konstruáljuk meg a $G_{i-1,j}$ gráfból a $G_{i,j}$ gráfot. Ezt úgy hajtjuk végre, hogy a W mátrixban a $G_{i-1,j}$ törölt éleinek megfelelő elemeknek ∞ értéket adunk. Ugyancsak végezzük el W-ben az algoritmus c) pontja szerint esedékes él-súly törléseket is a megfelelő értékek ∞-re állításával. Határozzuk meg $R_{i,j}$-t, a redukciók összegét, és redukáljuk $G_{i,j}$ súlyait. Számítástechnikailag ezt úgy vitelezzük ki, hogy előbb a W egy-egy sorában levő elemek mindegyikét csökkentjük a sor minimális elemének értékével, majd a W oszlopaiban levő elemek mindegyikét csökkentjük az oszlop minimális elemének értékével. A sor- és oszlopredukciók összege R_{ij}-t szolgáltatja. Ezután az alsó becslés:

$$A(H_{ij}) = \begin{cases} A(H_{i-1,j})+R_{ij}, & \text{ha} \quad R_{ij} < \infty, \\ \infty \quad \text{egyébként.} \end{cases}$$

Ha $A(H_{ij})=\infty$, akkor nyilván $H_{ij}=\emptyset$.

Legyen $\alpha(i+j)=0$

4. Lépés. Ha $A(H_{ij}) \geqq \text{OPT}$, akkor H_{ij} nem tartalmazhat az ε vektorban tároltnál jobb megoldást, ezt az ágat lezárjuk, és a 7. lépés következik.

Egyébként térjünk az 5. lépésre.

5. Lépés. Ha $i<|J|+1$, akkor még van választási lehetőség, és a becslés szerint még akadhat OPT-nál jobb értékű megoldás, folytassuk a 2. lépésnél.

Egyébként térjünk a 6. lépésre.

6. Lépés. E lépést az előzőek szerint csak $i=|J|+1$ és az $A(H_{ij})<\text{OPT}$ feltétel mellett hajtjuk végre. A feltétel teljesülése jelzi, hogy a meglevőnél jobb megoldást találunk. Legyen $\text{OPT}=A(H_{ij})$, és az $e_1, \ldots, e_{|J|+1}$ éleket tároljuk egy ε vektorban. Folytassuk az algoritmust a 7. lépésnél.

7. Lépés. Ha $A(H_{i-1,j+1})<\text{OPT}$, akkor $\alpha(i+j)=1$, $i:=i-1$ és $j:=j+1$ legyen. Állítsuk elő a G_{ij} gráfot, majd redukáljuk azt. Térjünk vissza a 2. lépéshez. Ha $A(H_{i-1,j+1}) \geqq \text{OPT}$, a 8. lépés következik.

8. Lépés. Ha $i+j=1$, akkor az ε vektorból állítsuk össze a minimális súlyú *Hamilton-kör*t, és az algoritmus befejeződött. Ha $i+j>1$, a 9. lépés következik.

9. Lépés. Ha $\alpha(i+j-1)=0$, akkor $i:=i-1$, és a 7. lépés következik. Ha $\alpha(i+j-1)=1$, akkor $j:=j-1$, és a 8. lépésre térünk át.

4. Egy numerikus példa

Az algoritmus működését egy numerikus példán keresztül mutatjuk be, részint a 3. pontban tömören leírt algoritmus jobb megértése kedvéért, részint a számítógépi kivitelezés bemutatásáért. A táblázatban, valamint a 2. ábrán feltüntetett b_i értékeken keresztül azt tartjuk számon, hogy a J_{AB} halmazhoz tartozó csúcsok az S'_A vagy S'_B halmazhoz sorolódnak. Például, ha egy b sorozatban $b_3=1$, akkor az $i=3$ és $b_3=0$-nak megfelelő sort és a $j=3$ és $b_3=0$-nak megfelelő oszlopot töröljük a mátrixból. Ha két csúcs között nincs irányított él, azt a táblázat megfelelő cellájában M-mel jelöltük. (M egy igen nagy numerikus értéknek feleltethető meg.)

A mintafeladatban

1. TÁBLÁZAT. A numerikus példa W mátrixának kezdeti alakja

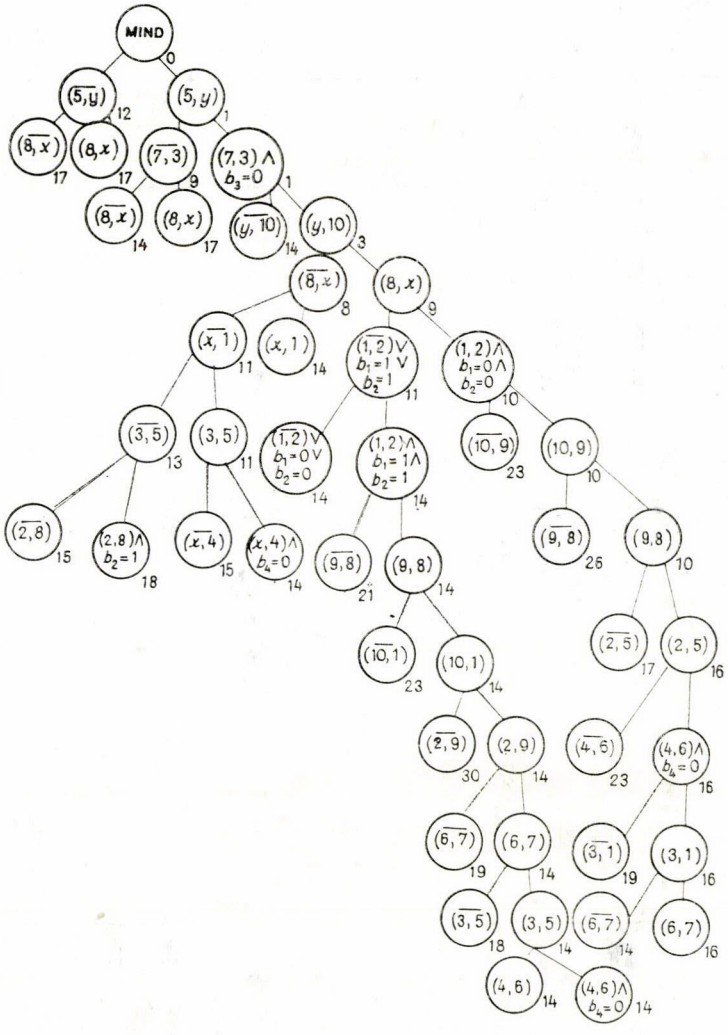
i \ j		1	1	2	2	3	3	4	4	5	6	7	8	9	10	x	y
	b_j	0	1	0	1	0	1	0	1	–	–	–	–	–	–	–	–
	b_i																
1	0	M	M	1	M	15	M	17	M	14	12	19	M	M	M	M	0
1	1	M	M	M	3	M	11	M	11	M	M	M	6	8	18	0	M
2	0	13	M	M	M	11	M	7	M	0	9	12	M	M	M	M	0
2	1	M	14	M	M	M	18	M	4	M	M	M	2	0	13	0	M
3	0	6	M	17	M	M	M	2	M	0	16	4	M	M	M	M	0
3	1	M	7	M	1	M	M	M	3	M	M	M	18	19	5	0	M
4	0	16	M	13	M	8	M	M	M	4	2	9	M	M	M	M	0
4	1	M	13	M	19	M	18	M	M	M	M	M	12	14	14	0	M
5	—	12	M	12	M	19	M	12	M	M	18	15	M	M	M	M	0
6	—	8	M	0	M	14	M	15	M	6	M	1	M	M	M	M	0
7	—	17	M	15	M	1	M	13	M	17	9	M	M	M	M	M	0
8	—	M	9	M	5	M	7	M	14	M	M	M	M	6	18	0	M
9	—	M	11	M	4	M	4	M	10	M	M	M	4	M	13	0	M
10	—	M	3	M	8	M	5	M	18	M	M	M	4	1	M	0	M
x	—	0	M	0	M	0	M	0	M	0	0	0	M	M	M	M	0
y	—	M	0	M	0	M	0	M	0	M	M	M	0	0	0	0	M

$$J_{AB} = \{1, 2, 3, 4\}, \quad J_A = \{5, 6, 7\}, \quad J_B = \{8, 9, 10\},$$

így $|V|=12$. A számítás menetét a 2. ábrán mutatjuk be. Az ábrán karikában feltüntettük az élre vonatkozó információkat, mely alapján a szétválasztás történik. Például, $(1, 2)\wedge b_1=0$, $b_2=0$ azt jelzi, hogy az 1 csúcsból a 2 csúcsba irányított él

alapján választunk szét, és mivel két ilyen él létezik G-ben, a $b_1=0 \wedge b_2=0$ azonosítja, hogy az A gépen történő beállításnak megfelelő élt választottuk ki. A karikában levő másik jelöléstípus például a $(4, 6) \wedge b_n=0$. Ez jelzi, hogy a (4, 6) élt tartalmazó *Hamilton-kör*ök csak olyan ütemezésnek feleltethetők meg, ahol a $4 \in J_{AB}$ munka az A gépre kerül, azaz $e \in S'_A$, és $b_n=0$ kell legyen. Ha egy b_i értéket rögzítünk ily módon, az $1-b_i$-nek megfelelő sor és oszlop minden eleme az M értéket kell hogy felvegye.

A karikán belül $(\bar{u}, \bar{v})$ jelzi, hogy a megfelelő részhalmaz az (u, v) élt nem tartalmazó *Hamilton-kör*ökből áll. A karikák mellett levő numerikus érték az alsó becslés.



2. ábra

A 2. ábrából látható, hogy az első megoldás az

$$\{x, 4, 6, 7, 3, 1, 2, 5, y, 10, 9, 8, x\}$$

Hamilton-kör, mely súlyainak összege 16. A második megoldás, mely optimális 14 értékű súlyösszeggel, az

$$\{x, 4, 6, 7, 3, 5, y, 10, 1, 2, 9, 8, x\}$$

Hamilton-kör. Ez az $S=(S_A, S_B)$ optimális ütemezésnek feleltethető meg, ahol

$$S_A = \{4, 6, 7, 3, 5\} \quad \text{és} \quad S_B = \{10, 1, 2, 9, 8\},$$

és a gépbeállítási idők összege szintén 14.

5. Végső megjegyzés

A fenti algoritmust HOLYINKA PÉTER kollégánk R20 számítógépre programozta, BASIC FORTRAN nyelven. Segítségéért ezúton szeretnénk köszönetet mondani.

IRODALOM

[1] LITTLE, J. D. C., MURTY, K. G., SWEENEY, D. W. and KAREL, C., "An algorithm for the traveling salesman problem", *Operations Research* **11** (1963) 972—989.

[2] SCHWARTZ, E. S., "*A heuristic procedure for parallel sequencing with choice of machines*", *Management Science* **10** (1964) 767—777.

(Beérkezett: 1975. november 15.)

HEPPES ALADÁR
SZÁMÍTÓGÉPALKALMAZÁSI KUTATÓ INTÉZET
1536 BUDAPEST I., CSALOGÁNY U. 30–32.
LUGOSI GÁBOR
KGM IPARGAZDASÁGI, SZERVEZÉSI ÉS SZÁMÍTÁSTECHNIKAI INTÉZET
1016 BUDAPEST I., KRISZTINA KRT. 55.

PRODUCTION SCHEDULING TO MINIMIZE SET-UP TIME OF ALTERNATIVE MACHINES

A. HEPPES and G. LUGOSI

An algorithm is presented for the following problem: A number of jobs is to be scheduled for a shop operating two machines. A subset of the jobs can be processed only on machine A and an other subset only on machine B while the rest of the jobs on both of the two machines. The objective of the scheduling is to minimize the sum of the set-up times.

In the graph formulation of the above problem the optimal schedule corresponds to the *Hamilton circle* of minimal weight in a special graph. For the determination of this *Hamilton circle* a branch-and-bound type algorithm has been developed. Numerical illustration of the algorithm is also presented.

Alkalmazott Matematikai Lapok **2** *(1976) 93—101*

A PRÉKOPA-FÉLE STABIL SZTOCHASZTIKUS PROGRAMOZÁSI MODELL NUMERIKUS MEGOLDÁSÁRÓL

SZÁNTAI TAMÁS
Budapest

A dolgozatban megmutatjuk, hogy a STABIL sztochasztikus programozási modellből származó nemlineáris programozási feladat megoldására sikerrel alkalmazható a *Veinott-féle metszősík algoritmus*. A modellben szereplő valószínűség értékek számítására egy új szimulációs eljárást alkalmazunk. Ennek, valamint az új nemlineáris prgramozási algoritmus használatának a hatására a számítási idő a [3] dolgozatban közöltnél lényegesen kisebbé válik. Ez lehetővé teszi a modell sztochasztikus jellegének egy részletesebb, a dolgozat utolsó részében leírt elemzését.

1. A Veinott-féle metszősík algoritmus alkalmazása sztochasztikus programozási feladatok megoldására

Tekintsük a következő nemlineáris programozási feladatot:

$$\min \mathbf{c}'\mathbf{x} \tag{1.1}$$

feltéve, hogy

$$g_1(\mathbf{x}) \leqq 0,\ g_2(\mathbf{x}) \leqq 0, \ldots, g_m(\mathbf{x}) \leqq 0.$$

Tegyük fel, hogy az esetleges lineáris és nemnegativitási feltételek a $g_i(\mathbf{x}) \leqq 0$, $i = 1, 2, \ldots, m$ feltételek között vannak felsorolva. Az (1.1) feladat megengedett megoldásainak a halmazát jelöljük D-vel, azaz legyen

$$D = \{\mathbf{x} : \mathbf{x} \in R^n,\ g_i(\mathbf{x}) \leqq 0,\ i = 1, 2, \ldots, m\}. \tag{1.2}$$

Az (1.1) feladatban szereplő nemlineáris függvényekre a következő megszorításokat tesszük.

(i) A megengedett megoldások D halmaza befoglalható kell hogy legyen egy lineáris egyenlőtlenség-rendszer által definiált, Z^1 korlátos konvex poliéderbe. (Feltehetjük, hogy a feladat feltételei között esetleg szereplő lineáris és nemnegativitási feltételeket mindig bevesszük a Z^1 korlátos konvex poliédert definiáló lineáris egyenlőtlenségek közé.)

(ii) A $g_i(\mathbf{x})$, $i = 1, 2, \ldots, m$ feltételi függvények legyenek kvázi-konvexek a Z^1 poliéderen.

(iii) Létezzen legalább egy olyan $\mathbf{x}^0 \in D$ pont, amelyre $g_i(\mathbf{x}^0) < 0$ minden olyan $1 \leqq i \leqq m$ indexre, amelyre a $g_i(\mathbf{x})$ függvény nemlineáris.

Megjegyezzük, hogy az (1.1) nemlineáris programozási feladat célfüggvényére is elég lenne csak a konvexitás, illetve kvázikonvexitás megkövetelése ahhoz, hogy a feladat megoldására a *Veinott-féle metszősík algoritmus* alkalmazható legyen (lásd [5], illetve [2]). Tekintettel azonban arra, hogy a konvex célfüggvény esete

egyszerűen visszavezethető a lineáris célfüggvény esetére (lásd pl. [1]), valamint hogy a STABIL sztochasztikus programozási modellből származó nemlineáris programozási feladatban is lineáris a célfüggvény, itt csak az (1.1) alakú feladat megoldó algoritmusát írjuk le. Ez a következő (lásd [2]):

I. Fázis. Keressünk egy $\mathbf{x}^0 \in D$ pontot, amely eleget tesz az (iii) követelménynek.

II. Fázis. Legyen $r=1$ kezdetben, és válasszunk egy $\varepsilon>0$ tűrést.

1. Lépés. Keressük meg az

$$\min \mathbf{c}'\mathbf{x} \tag{1.3}$$

feltéve, hogy

$$\mathbf{x} \in Z^r$$

lineáris programozási feladat egy optimális megoldását, amelyet jelöljön $\mathbf{z}^r$. Ha $\mathbf{z}^r \in D$, akkor $\mathbf{z}^r$ az (1.1) feladat optimális megoldása, és az algoritmus befejeződik. Ellenkező esetben menjünk a 2. lépésre.

2. Lépés. Keressük meg azt a legnagyobb $0 \leqq \lambda \leqq 1$ valós számot, amelyre még $\mathbf{x}^0 + \lambda(\mathbf{z}^r - \mathbf{x}^0) \in D$. Legyen ez a szám λ_r, és vezessük be az $\mathbf{x}^r = \mathbf{x}^0 + \lambda_r(\mathbf{z}^r - \mathbf{x}^0)$ jelölést.

a) Ha $\mathbf{c}'\mathbf{x}^r - \mathbf{c}'\mathbf{z}^r \leqq \varepsilon$, akkor $\mathbf{x}^r$ az (1.1) feladat egy közelítőleg optimális megoldása (a célfüggvény értéke az optimum értéktől ε-nál kevesebbel tér el), és az algoritmus befejeződik.

b) Ellenkező esetben válasszunk egy olyan i_r indexet, amelyre $g_{i_r}(\mathbf{x}^r) = 0$ (több lehetőség esetén a választás tetszőleges lehet). Legyen

$$Z^{r+1} = \{\mathbf{x} : \mathbf{x} \in Z^r, \nabla g_{i_r}(\mathbf{x}^r)(\mathbf{x} - \mathbf{x}^r) \leqq 0\},$$

és r értékét eggyel növelve menjünk az 1. lépésre.

Az algoritmus 2. lépésében megtehetjük, hogy azt a legnagyobb $0 \leqq \lambda \leqq 1$ valós számot keressük, amelyre még $\mathbf{x}^{-r} + \lambda(\mathbf{z}^r - \mathbf{x}^{-r}) \in D$, és ha ez a szám λ_r, akkor az $\mathbf{x}^r = \mathbf{x}^{-r} + \lambda_r(\mathbf{z}^r - \mathbf{x}^{-r})$ pont mellett az $\mathbf{x}^{-r-1} = \mathbf{x}^{-r} + q(\mathbf{x}^r - \mathbf{x}^{-r})$ pontot is képezzük, ahol $0 < q < 1$ egy rögzített valós szám. Így az $\mathbf{x}^0$ belső pont helyett megengedett pontok egy $\mathbf{x}^{-1}, \mathbf{x}^{-2}, \ldots, \mathbf{x}^{-r}, \ldots$ sorozatát használjuk, amelyekről bizonyítható, hogy az $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^r, \ldots$ sorozathoz hasonlóan az (1.1) feladat optimális megoldásához konvergálnak. Az így keletkező algoritmus azonos ZOUTENDIJK *módosított megengedett irányok módszere* algoritmusával (lásd [6] és [7]). Sajnos a módosított algoritmus konvergenciáját csak konvex feltételi függvények esetére bizonyította be ZOUTENDIJK, és a számítási tapasztalatok is azt mutatják, hogy $q = \frac{1}{2}$ választással, kvázi-konvex feltételi függvények esetén az algoritmus téves eredményre vezethet. A problémát feltehetően az okozza, hogy az $\mathbf{x}^{-1}, \mathbf{x}^{-2}, \ldots, \mathbf{x}^{-r}, \ldots$ pontsorozat „túl gyorsan" közelít a D halmaz határához.

A STABIL sztochasztikus programozási modellből származó nemlineáris programozási feladat megoldására a fent leírt módosított algoritmust alkalmaztuk azzal a különbséggel, hogy az $\mathbf{x}^{-r}$, $r=2, 3, \ldots$ sorozatot az $\mathbf{x}^{-r-1} = \mathbf{x}^{-r} + \frac{1}{r+1}(\mathbf{x}^r - \mathbf{x}^{-r})$, $r=1, 2, \ldots$ rekurzív összefüggéssel definiáltuk. Megjegyezzük, hogy kvázikonvex

feltételi függvényeket tartalmazó nemlineáris programozási feladat esetén az így keletkező algoritmusra sem ismert konvergencia bizonyítás. Tekintettel azonban arra, hogy a sztochasztikus programozási feladatok megoldásakor általában nem törekszünk (a valószínűség értékek viszonylag pontatlan számítása miatt nem is törekedhetünk) az optimális megoldás igen pontos közelítésére, a konvergencia bizonyítása inkább elméleti, mint gyakorlati jelentőséggel bír csak. A számítási tapasztalatok azt mutatják, hogy az $\mathbf{x}^{-r}$, $r=2, 3, \ldots$ sorozat ily módon történő képzésével elég gyorsan el lehet jutni az optimális megoldás közelébe (ez a STABIL sztochasztikus programozási modell esetén 5—10 iterációt jelent).

A STABIL sztochasztikus programozási modellből származó nemlineáris programozási feladat a következő alakú:

$$\min \mathbf{c}'\mathbf{x} \tag{1.4}$$

feltéve, hogy

$$g(\mathbf{x}) = P\left(\frac{1}{\sigma_i}(\mathbf{a}_i'\mathbf{x} - b_i) \geqq \beta_i,\ i = 1, 2, \ldots, m\right) \geqq p,$$

$$\mathbf{a}_i'\mathbf{x} \geqq b_i, \quad i = m+1, \ldots, m+M,$$

ahol $\mathbf{x} \in R$ (lásd [3]). A megoldott feladatokban $m=4$, $M=129$ volt, és a $\beta_1, \beta_2, \beta_3, \beta_4$ valószínűségi változók különböző korreláció mátrixszal bíró, standard normális eloszlásúak voltak. (Megjegyezzük, hogy a 129 darab lineáris feltétel között 23 olyan feltétel pár található, amelyek lineáris egyenlőség feltételeket reprezentálnak, 12 feltétel egyedi felső korlátokat, 46 feltétel pedig egyedi alsó korlátokat, azaz nemnegativitási feltételeket ír le.)

Erre a feladatra a *Veinott-féle metszősík algoritmus* első fázisának a problémáját igen egyszerűen lehetett megoldani. Elegendő volt ugyanis az

$$\max \sum_{i=1}^{m} \frac{1}{\sigma_i}(\mathbf{a}_i'\mathbf{x} - b_i) \tag{1.5}$$

feltéve, hogy

$$\mathbf{a}_i'\mathbf{x} \geqq b_i, \quad i = m+1, \ldots, m+M$$

lineáris programozási feladatot megoldani. Az (1.5) feladat optimális megoldásvektorára a valószínűségi szint 0,99 körüli értéknek adódott. Megjegyezzük, hogy amennyiben ez az eljárás nem vezetne eredményre, az első fázis feladata megoldható lenne a *Veinott-féle metszősík algoritmus* második fázisának egy alkalmasan módosított feladatra történő alkalmazásával.

Ugyancsak könnyen meg lehetett adni egy, az (1.4) feladat megengedett megoldásait magába foglaló kezdeti Z^1 korlátos konvex poliédert. Erre a célra ugyanis megfelel az

$$\mathbf{a}_i'\mathbf{x} \geqq b_i, \quad i = m+1, \ldots, m+M \tag{1.6}$$

lineáris egyenlőtlenségek által leírt konvex poliéder.

A második fázis végrehajtása során a fő nehézséget a λ_r szám meghatározása és a $\nabla g(\mathbf{x}^r)$ gradiens vektor kiszámítása okozta. Mindkét esetben a szimulációs módszerrel számított valószínűség értékekben fellépő pontatlanságok hatása ellen kellett védekezni. Ezt egyrészt a szimulációs kiértékelés tökéletesítésével értük el,

amelyet vázlatosan a dolgozat 2. szakaszában ismertetünk, másrészt maga a *Veinott-féle metszősík algoritmus* is jó lehetőséget nyújt az esetleges számítási pontatlanságok elleni védekezésre. A II. fázis 2. lépésében λ_r meghatározása történhet például az $\mathbf{x}^0$ (illetve $\mathbf{x}^{-r}$) és $\mathbf{z}^r$ pontokat összekötő szakaszból kiinduló, intervallum felező algoritmussal. Ennek során nem szükséges a λ_r érték egészen pontos meghatározása, elegendő helyette egy, a pontos értéket jól közelítő felső korlátot megadni. Ennek hatására a metszések ugyan kevésbé lesznek hatékonyak, de mindenképpen megbízhatóbbak abban az értelemben, hogy a gradiens vektor pontatlan számítása miatt nem fognak a D halmazba bemetszeni. Az iterációk számának a növekedtével azután az egymásutáni metszéseket egyre közelebb lehet engedni a D halmaz határához, és ugyanakkor növelni kell a szimulációs kiértékelés pontosságát. Ily módon eljárva a *Veinott-féle metszősík algoritmus* alkalmazható volt olyan sztochasztikus programozási modellekből származó nemlineáris programozási feladatok megoldására is, amelyekben a nemlineáris feltételi függvény parciális deriváltjait csak numerikusan lehetett közelíteni (például a többdimenziós gamma eloszlás esete).

2. Szimulációs eljárás a többdimenziós normális eloszlásfüggvény és gradiense értékeinek meghatározására

Legyenek ξ_i, $i=1, 2, \ldots, n$ nulla várható értékű, egy szórású és $\mathbf{R}$ korreláció mátrixú valószínűségi változók. A

$$P(\xi_i \leqq x_i, \quad i = 1, 2, \ldots, n) \tag{2.1}$$

valószínűség értékének a meghatározása szimulációs módszerrel a legegyszerűbben úgy történhet, hogy előállítunk egy, a $\xi_1, \xi_2, \ldots, \xi_n$ valószínűségi változók eloszlásának megfelelő, véletlen számokból álló vektorsorozatot, és leszámláljuk, hogy a sorozat hány elemére teljesül a (2.1) valószínűségben szereplő feltételek mindegyike. Az így nyert számot a véletlen vektorsorozat hosszával osztva egy relatív gyakoriság értéket nyerünk, amely elég hosszú sorozat használata esetén a (2.1) valószínűséget kielégítő pontossággal közelíti.

A (2.1) valószínűség az általános valószínűségi tétel segítségével a következőképpen alakítható át:

$$P(\xi_i \leqq x_i, \quad i = 1, 2, \ldots, n) = D(x_1, x_2, \ldots, x_n) + H(x_1, x_2, \ldots, x_n), \tag{2.2}$$

ahol

$$D(x_1, x_2, \ldots, x_n) = 1 - n + \sum_{i=1}^{n} \Phi(x_i),$$

$$H(x_1, x_2, \ldots, x_n) = \sum_{1 \leqq i < j \leqq n} P(\xi_i \geqq x_i, \xi_j \geqq x_j) -$$

$$- \sum_{1 \leqq i < j < k \leqq n} P(\xi_i \geqq x_i, \xi_j \geqq x_j, \xi_k \geqq x_k) + \ldots + (-1)^n P(\xi_i \geqq x_i, \ i = 1, \ldots, n),$$

és $\Phi(x)$ az egydimenziós standard normális eloszlás eloszlásfüggvénye.

A (2.2) átalakítás előnye az, hogy a jobboldalon szereplő $D(x_1, x_2, ..., x_n)$ determinisztikusan számítható kifejezés értéke különösen az egyhez közeli valószínűségek számításakor könnyen dominálhatja a teljes jobboldali értéket, mely által a $H(x_1, x_2, ..., x_n)$ kifejezés szimulációs módszerrel történő meghatározása lényegesen kisebb hibára vezet. Egy, a $H(x_1, x_2, ..., x_n)$ kifejezés szimulációs kiértékelésére szolgáló hatékony algoritmust a [4] dolgozatban írtunk le, és ugyanitt találhatók meg a módszerrel kapcsolatos első számítási tapasztalatok is.

A leírt módszer alkalmazható a többdimenziós normális eloszlásfüggvény gradiense értékeinek a meghatározására is, ugyanis az egyes parciális deriváltak számítása egyszerűen visszavezethető egy eggyel alacsonyabb dimenziójú többdimenziós normális eloszlásfüggvény értékének a számítására (lásd [3]).

3. Számítási eredmények

A számításokat UNIVAC 1108 típusú számítógépen végeztük (amely lényegesen gyorsabb, mint a CDC 3300 típusú számítógépek), így az időeredmények csak nehezen hasonlíthatók össze a [3] dolgozatban közöltekkel. A célunk azonban nem is az összehasonlítás, hanem annak megmutatása, hogy nagy teljesítményű számítógépen, magas szintű felhasználói software alkalmazásával a STABIL sztochasztikus programozási feladat megoldása rutin jelleggel történhet.

A feladatok megoldása során használtuk a UNIVAC 1108 számítógépekre a SCICON *Computer Services* által kifejlesztett SCICONIC nevű matematikai programozási programrendszert, valamint egy standard normális eloszlású véletlen szám generáló szubrutint, amelyet MARSAGLIA, és BRAY módszere alapján M. C. PEARCE készített az *Imperial College* matematikai tanszékén. Tekintettel arra, hogy a SCICONIC matematikai programozási programrendszer nem tartalmaz duál szimplex algoritmust megvalósító programokat, az egyes iterációk során mindig az (1.3) lineáris programozási feladat duálját oldottuk meg. Így az új metszősík bevezetése a duális feladatban egy új változó bevezetésével volt ekvivalens. Sajnos a programrendszer lineáris programozási feladatokat megoldó algoritmusát sem lehetett szubrutinként használni, így minden egyes iteráció során újra kellett generálni az új metszősíknak megfelelő változóval kibővített duál lineáris programozási feladatot, és a metszősík algoritmus iterációit egy, a job-control nyelven belüli, meglehetősen nehézkesen felépíthető ciklussal lehetett csak megvalósítani. Nyilvánvaló, hogy a metszősík algoritmus minden iterációja során jelentős gépidő megtakarítás érhető el, ha olyan lineáris programozási programrendszert alkalmazunk, amelyben nem szükséges a teljes feladat újraelőállítása (csak az új változónak megfelelő adatoké), és amelyben az új lineáris programozási feladat megoldása a régi optimális megoldásából kiindulva folytatható. További lényeges időmegtakarítást jelenthet, ha a standard normális eloszlású véletlen szám generáló szubrutin bizonyos részeit gépi kódban írjuk meg, amint azt MARSAGLIA és társa is javasolja. Mindezek mellett az egyes feladatok megoldása során nem a gyorsaságra, hanem a biztonságra törekedtünk, így végig 1000 hosszúságú véletlen vektorsorozatokra végeztük a szimulációs kiértékeléseket, ami a metszősík algoritmus első iterációiban túlzottan pontos számításokra vezetett. A fentiek figyelembevételével azt lehet állítani, hogy a jelen dolgozatban közölt időeredmények is könnyen tovább javíthatók.

A STABIL sztochasztikus programozási modellből származó feladatot a [3] dolgozatban közölt adatokkal oldottuk meg. Az előírt valószínűségi szintnek és a korreláció mátrixnak megfelelően három esetet különböztettünk meg:

$$1.\ eset:\ p = 0{,}9;\ \mathbf{R} = \begin{pmatrix} 1{,}0 & -0{,}8 & 0{,}4 & 0{,}4 \\ -0{,}8 & 1{,}0 & 0{,}1 & 0{,}1 \\ 0{,}4 & 0{,}1 & 1{,}0 & 0{,}9 \\ 0{,}4 & 0{,}1 & 0{,}9 & 1{,}0 \end{pmatrix},$$

$$2.\ eset:\ p = 0{,}95;\ \mathbf{R} = \begin{pmatrix} 1{,}0 & -0{,}8 & 0{,}4 & 0{,}4 \\ -0{,}8 & 1{,}0 & 0{,}1 & 0{,}1 \\ 0{,}4 & 0{,}1 & 1{,}0 & 0{,}9 \\ 0{,}4 & 0{,}1 & 0{,}9 & 1{,}0 \end{pmatrix},$$

$$3.\ eset:\ p = 0{,}9;\ \mathbf{R} = \begin{pmatrix} 1{,}0 & -0{,}7 & 0{,}3 & 0{,}3 \\ -0{,}7 & 1{,}0 & 0{,}1 & 0{,}1 \\ 0{,}3 & 0{,}1 & 1{,}0 & 0{,}9 \\ 0{,}3 & 0{,}1 & 0{,}9 & 1{,}0 \end{pmatrix}.$$

Az 1. táblázat a kiinduló Z^1 poliéderre vonatkozó determinisztikus feladat optimális megoldását, valamint a három esetnek megfelelő sztochasztikus programozási feladat optimális megoldásait tartalmazza. 3. esetben eltérés mutatkozik a [3] dolgozatban közölt eredményektől. Az általunk nyert optimális megoldás ugyanis azt mutatja, hogy a korreláció mátrix bizonyos elemeinek a kismértékű megváltoztatása nem befolyásolja lényegesen a célfüggvény optimális értékét. Érdemes megvizsgálni, hogy minek tulajdonítható ez a különben nem várt jelenség. Ha kiszámítjuk a sztochasztikus feltételek baloldali értékeit a háromesetnek megfelelő optimális megoldásokra, akkor azt találjuk, hogy az előírt valószínűségek a következőképpen realizálódnak:

$1.\ eset:\ P(2{,}238 \geqq \beta_1;\ 7{,}180 \geqq \beta_2;\ 11{,}962 \geqq \beta_3;\ 1{,}333 \geqq \beta_4) \sim 0{,}908,$

$2.\ eset:\ P(2{,}238 \geqq \beta_1;\ 3{,}987 \geqq \beta_2;\ 11{,}969 \geqq \beta_3;\ 1{,}641 \geqq \beta_4) \sim 0{,}948,$

$3.\ eset:\ P(2{,}238 \geqq \beta_1;\ 2{,}332 \geqq \beta_2;\ 11{,}969 \geqq \beta_3;\ 1{,}344 \geqq \beta_4) \sim 0{,}901.$

Az 1. és 2. eset összehasonlításából az olvasható le, hogy a megbízhatósági szint 0,9 fölé emeléséhez a β_4 valószínűségi változó felső korlátját kell feltétlen növelni (emellett β_2 felső korlátja még csökkenthető is). Az 1. és 3. eset összevetése pedig azt mutatja, hogy a korreláció mátrix elemeinek a változtatása csak a β_2 valószínűségi változó felső korlátjára van hatással. β_2 felső korlátját pedig egyedül az x_{26} változó szabályozza, mégpedig oly módon, hogy x_{26} viszonylag kis változásai a felső korlátban lényeges változásokat hoznak létre. Ugyanakkor x_{26} változásai a célfüggvény értékében csak kis változást eredményeznek, és ez magyarázza a nyert eredményeket. A részletesebb közgazdasági elemzés igénye nélkül megjegyezzük, hogy az x_{26} változó a villamosenergia-ipari ágazati tőkés import értéket képviseli, ezért a nyert eredmények azt látszanak kifejezni, hogy egy megoldás annál megbízhatóbb, minél

1. TÁBLÁZAT. Az optimális mogoldások

Komponens sorszáma	Determinisztikus feladat	1. eset feladata	2. eset feladata	3. eset feladata
1	6 202,43	6 442,59	6 496,53	6 443,24
2	12 756,90	12 808,67	12 820,30	12 808,81
3	4 640,00	4 640,00	4 640,00	4 640,00
4	8 116,90	8 168,67	8 180,30	8 168,81
5	0,00	0,00	0,00	0,00
6	4 460,00	4 460,00	4 460,00	4 460,00
7	42,89	40,00	41,29	41,95
8	4 730,91	4 562,51	4 523,57	4 560,92
9	0,68	0,69	0,69	0,69
10	0,00	0,00	0,00	0,00
11	0,00	0,00	0,00	0,00
12	0,00	0,25	0,25	0,25
13	0,60	0,40	0,42	0,40
14	9 134,00	9 134,00	9 134,00	9 134,00
15	568,00	568,00	568,00	568,00
16	1 327,00	1 327,00	1 327,00	1 327,00
17	498,39	498,56	498,60	498,56
18	0,00	0,00	0,00	0,00
19	8 925,52	9 338,51	9 338,51	9 338,51
20	1 266,20	0,00	0,00	0,00
21	0,00	1 007,02	1 010,05	1 006,66
22	683,80	1 950,00	1 950,00	1 950,00
23	1 600,00	506,10	509,25	506,14
24	18 400,00	18 400,00	18 400,00	18 400,00
25	1 889,26	1 889,26	1 889,26	1 889,26
26	25,55	23,83	24,60	24,99
27	8 077,60	8 077,60	8 077,60	8 077,60
28	965,97	966,65	966,80	966,65
29	241,49	241,66	241,70	241,66
30	56 983,43	57 024,68	57 033,94	57 024,79
31	2 564,25	2 566,11	2 566,53	2 566,12
32	1 424,59	1 425,62	1 425,85	1 425,62
33	993,60	993,60	993,60	993,60
34	110,40	110,40	110,40	110,40
35	93,70	93,76	93,78	93,76
36	4 467,50	4 463,94	4 463,14	4 463,93
37	518,39	518,56	518,60	518,56
38	116,53	115,85	115,70	115,85
39	3 224,31	3 224,25	3 224,23	3 224,25
40	1 914,81	1 913,09	1 913,86	1 914,25
41	964,52	887,19	906,25	887,54
42	2 258,66	2 121,33	2 121,52	2 121,34
43	2 720,02	2 483,81	2 481,51	2 483,78
44	2 123,37	3 175,22	3 172,11	3 175,18
45	3 358,95	2 680,64	2 685,87	2 680,70
46	0,00	0,00	0,00	0,00
Valószínűségi szint	0,271	0,908	0,948	0,901
Célfüggvény érték	4 373,80	4 370,17	4 369,36	3 470,16

kisebb tőkés import értéket enged meg. Valószínű, hogy a tőkés import értékének a minimalizálását a modellbe beépítve az előbb leért jelenség is megszűnne.

A 2., 3. és 4. táblázatok rendre az 1., 2. és 3. eset futási eredményeinek a rövid összefoglalását tartalmazzák. A táblázatokban az első iteráció eredményei mindig a kiinduló Z^1 korlátos konvex poliéderen történő optimalizálás eredményeit, a második interáció eredményei pedig az (1.5) probléma megoldásának eredményeit jelentik. Ezek után a tényleges metszősík algoritmus iterációs eredményei következnek. A közölt időeredmények mind másodpercben értendők.

2. TÁBLÁZAT Az első feladat iterációs eredményei

Iteráció sorszáma	Valószínűségi szint	Célfüggvény érték	Mátrix generálás ideje	Szimplex algoritmus ideje	Szimulációs idő	Összes idő
1	0,271	4373,80	2,86	2,49	—	5,35
2	0,994	4206,94	2,49	1,67	—	4,16
3	0,875	4370,65	3,61	1,68	6,92	12,21
4	0,003	4370,45	3,45	1,64	5,87	10,96
5	0,848	4370,35	4,20	1,72	11,63	17,55
6	0,897	4370,34	4,57	1,69	8,44	14,70
7	0,908	4370,17	4,90	1,72	10,72	17,34
Összesen	—	—	26,08	12,61	43,58	82,27

3. TÁBLÁZAT. A második feladat iterációs eredményei

Iteráció sorszáma	Valószínűségi szint	Célfüggvény érték	Mátrix generálás ideje	Szimplex algoritmus ideje	Szimulációs idő	Összes idő
1	0,271	4373,80	2,89	2,43	—	5,32
2	0,994	4206,94	2,63	1,74	—	4,37
3	0,000	4369,46	3,53	1,68	6,23	11,44
4	0,928	4369,37	3,81	1,73	5,94	11,48
5	0,947	4369,36	4,21	1,69	7,83	13,73
6	0,947	4369,36	4,63	1,74	8,12	14,49
7	0,948	4369,36	4,38	1,72	10,23	16,33
Összesen	—	—	26,08	12,73	38,35	77,16

4. TÁBLÁZAT. A harmadik feladat iterációs eredményei

Iteráció sorszáma	Valószínűségi szint	Célfüggvény érték	Mátrix generálás ideje	Szimplex algoritmus ideje	Szimulációs idő	Összes idő
1	0,271	4373,80	2,92	2,36	—	5,28
2	0,994	4206,94	2,82	1,73	—	4,55
3	0,869	4370,74	3,21	1,64	6,64	11,49
4	0,000	4370,26	3,54	1,63	5,25	10,42
5	0,867	4370,17	3,70	1,83	6,65	12,18
6	0,901	4370,16	4,03	1,72	6,11	11,86
Összesen	—	—	20,22	10,91	24,65	55,78

IRODALOM

[1] KELLEY, J. E. JR., "The cutting-plane method for solving convex programs", *Journal of the SIAM* **8** (1960) 703—712.
[2] MARTOS, B., *Nonlinear Programming* (Akadémiai Kiadó, Budapest, 1975).
[3] PRÉKOPA, A., GANCZER, S., DEÁK, I. és PATYI, K., „A STABIL sztohasztikus programozási modell és annak kísérleti alkalmazása a magyar villamosenergia-iparra", *Alk. Mat. Lapok* **1** (1975) 3—22.
[4] SZÁNTAI, T., „Egy eljárás a többdimenziós normális eloszlásfüggvény és gradiense értékeinek meghatározására", *Alk. Mat. Lapok* **2** (1976).
[5] VEINOTT, A. F. JR., "The supporting hyperplane method for unimodal programming", *Operations Research* **15** (1967) 147—152.
[6] ZOUTENDIJK, G., "Nonlinear programming, computational methods", in: *Nonlinear programming* Ed. J. Abadie (North-Holland, Amsterdam, 1971) 37—86.
[7] ZOUTENDIJK, G., "Nonlinear programming: a numerical survey", *Journal of the SIAM Control* **4** (1966) 194—210.

(Beérkezett: 1976. május 20.)

SZÁNTAI TAMÁS
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1502 BUDAPEST XI., KENDE U. 13—17

ON NUMERICAL SOLUTION OF THE STABIL STOCHASTIC PROGRAMMING MODEL OF PRÉKOPA

T. SZÁNTAI

In the paper we show that the supporting hyperplane method of Veinott is a good one for solving nonlinear programming problems which arise from the STABIL stochastic programming model. We use a new simulation technique for calculating the necessary probability values. Applying these technics we get good computing times. We give a more detailed description of the stochastic properties of STABIL model, too.

Alkalmazott Matematikai Lapok **2** *(1976) 103—114*

A SZIMULTÁN TANULÁS DINAMIKAI ELMÉLETE

FARKAS MIKLÓS
Budapest

Determinisztikus, dinamikai modellt konstruálunk a tanulás időbeli folyamatának leírására abban az esetben, amikor a diák huzamos időn át, egyszerre több tárgyat tanul. A modell lineáris és strukturálisan stabilis, igy viszonylag könnyen felhasználható általános, kvalitaív következtetések levonására. A főbb eredmények a következők.Több tárgy párhuzamos tanulásával nagyobb intenzitást lehet elérni, mint egyetlen tárgy tanulásával. Az intenzitás mértéke számszerűen kifejezhető. Másképpen alakul a tanulás intenzitása az időben akkor, ha a tanulás „szabadon" és másképpen, ha rendszeres külső ellenőrzés mellett folyik. Kiadódik, hogy minél nagyobb a rendszeres időközönkénti ellenőrzés gyakorisága az egyik tárgyban, annál kevésbé zavarja ez a másik tárgy tanulását.

1. Bevezetés

Ebben a dolgozatban matematikai modellt javaslunk a tanulás időbeli folyamatának leírására abban az esetben, amikor a diák (iskolai, vagy egyéni tanuló, egyetemi hallgató, valamely továbbképző tanfolyam hallgatója stb.) hosszabb időn át, egyszerre több tárgyat tanul. Nem a tanulás során az emberi agyban lejátszódó folyamatokkal, a tanulás belső, lélektani, szellemi dinamikájával foglalkozunk (v.ö. [3]), hanem ennek csupán külső megnyilvánulásával az elsajátított anyagmennyiséggel mint az idő függvényével, illetve a tanulás intenzitásának változásával az időben. Ennek során ugyan bizonyos belső, szubjektív, pszichológiai tényezőket figyelembe kell vennünk, ezeket azonban a diákra jellemző állandóknak tekintjük.

A javasolt modell lineáris és durva (a durva szót itt nemcsak köznapi, hanem matematikai értelemben is használjuk az orosz грубая система kifejezés megfelelőjeként). Ennek előnye az, hogy jól kezelhető, strukturálisan stabilis és alkalmas a folyamat leglényegesebb, kvalitatív jellemzőinek bemutatására. Hátránya, hogy a finomabb, árnyaltabb leírásra nem alkalmas, és kvantitatív következtetéseket csak nagy óvatossággal vonhatunk le belőle. Úgy tűnik azonban, hogy tág lehetőségek kínálkoznak a további finomításra és pontosításra.

Az itt ismertetésre kerülő elméletben a diák modellje egy autonóm, illetve egy külső kényszernek alávetett dinamikai rendszer. Talán szükségtelen, de esetleges félremagyarázások megelőzésére mégis célszerű annak hangsúlyozása, hogy a matematikai modell sohasem azonosítható az általa leírt valóságos jelenséggel. A diák teljes ember és mint ilyen társadalmi, szellemi és biológiai lény, akinek számtalan lényeges tulajdonságától, jellemzőjétől eltekintettünk, mikor absztrakció útján, a szóban forgó jelenség vizsgálata érdekében modelljét megalkottuk. Ha valaha kísérlet történik a teljes ember leírására dinamikai rendszer segítségével, a keletkező modell távolról sem lesz ilyen egyszerű és sohasem lesz azonos magával az emberrel.

2. A tanulás folyamatát jellemző függvények és paraméterek

Jelöljük t-vel az időt, melyet valamely $t=0$ időpillanattól, a tanulmányok megkezdésének időpontjától mérünk. A t változó szerinti deriválást ponttal fogjuk jelölni. Tételezzük fel, hogy a $t=0$ időponttól kezdve huzamos időn át (ezen pl. egy egyetemi félévet, vagy egy tanévet értünk) a diáknak $n \geqq 2$ számú tárgyat kell tanulnia. Jelöljük x_k-val a k-adik tárgyból megtanult anyagmennyiséget, más szóval a diák *tudásmennyiségét* a k-adik tárgyban. A tudásmennyiséget például könyvoldalakban mérhetjük és nem foglalkozunk azzal a kérdéssel, hogy milyen mélységű a tudás, alkalmazni tudja-e a diák a tanultakat stb. Az x_k mennyiségekből megalkotjuk az $x=\operatorname{col}(x_1, x_2, \ldots, x_n)$ oszlopvektort, melyet *tudásmennyiség vektornak* nevezünk. Ami minket érdekel, az a tudásmennyiség vektor mint az idő függvénye:

$$x(t) = \operatorname{col}\left(x_1(t), x_2(t), \ldots, x_n(t)\right)$$

a diák tudásmennyisége a t időpontban. Az $x(t)$ függvényt vizsgálhatjuk a diák teljes élete folyamán („jó pap holtig tanul"), melyet gyakorlatilag a $[0, \infty)$ időintervallumnak tekintünk, ill. valamely $[0, T]$ intervallumban, ahol $T>0$ pl. az összes tárgyból egyszerre teendő vizsga időpontja, vagy a vizsgaidőszak kezdete.

Az $x(t)$ függvény deriváltját, az $\dot{x}(t)$ vektort a tanulás t időpontbeli *intenzitásának* nevezzük és a továbbiakban elsősorban ezzel foglalkozunk. Ha ugyanis az intenzitás vektor mint az idő függvénye ismeretes, a tudásmennyiség egyszerű integrálással adódik. Az $\dot{x}(t)$ intenzitás vektor durván szólva a tudásmennyiség megváltozása időegység alatt. Időegységnek célszerű egy napot, vagy egy hetet választani és gondolni kell arra, hogy a diák az időegységet nemcsak tanulással tölti. Ha a diák a t időpontban a k-adik tárgyat tanulja, akkor $\dot{x}_k(t)$ rendszerint (nem mindig!) pozitív. Ha a t időpontban az i-edik tárgyat nem tanulja, akkor $\dot{x}_i(t)$ általában nem zérus, hanem negatív szám, ui. a diák felejt. A felejtés bonyolult folyamatával a továbbiakban nem foglalkozunk külön. A tanulás intenzitásának nevezett $\dot{x}(t)$ vektor valójában a tudásmennyiség megváltozásának intenzitása, mely a tanulás és a felejtés egymással ellentétes komponenseiből tevődik össze. Az intenzitás vektor koordinátáinak „fizikai" dimenziója például könyvoldal per nap.

Szerepeltetni fogjuk az $x(t)$ függvény második deriváltját. Az $\ddot{x}(t)$ vektort a *tudás gyorsulásának* nevezzük. Ha $\ddot{x}_k(t)$ pozitív, akkor a t időpontban a k-adik tárgy tanulásának intenzitása nő, ha $\ddot{x}_k(t)$ negatív, akkor az intenzitás csökken.

A diákról feltételezzük, hogy akar tanulni, vagyis a tanulásra rendelkezésre álló időt teljes egészében tanulásra fordítja, munkabíró képessége és tehetsége az időben állandó, több időegységből (napból, esetleg hétből) álló időintervallum folyamán nem fárad ki, mivel a nap (hét) megfelelő hányadát a fizikai szükségletek kielégítésére és pihenésre, sportolásra stb. fordítja.

Bevezetjük és $b=\operatorname{col}(b_1, b_2, \ldots, b_n)$-nel jelöljük a diák *terhelhetőségi vektorának* fogalmát. A b oszlopvektor i-edik koordinátája b_i a diák *terhelhetősége az i-edik tárgyban.* Ezen a következőt értjük: Ha a diák egyedül az i-edik tárgyat tanulná, akkor tanulásának intenzitása b_i lenne. Más szóval b_i az a maximális (tovább már nem fokozható) intenzitás, melyet a diák az i-edik tárgyban ki tud fejteni, ha a többi tárgyat nem tanulja, de nem is felejti (a többi tárgy tanulásának intenzitása zérus). A b_i terhelhetőség értéke nagy, ha a tárgy az adott diáknak könnyű, vagyis b_i értéke a tárgytól és a diák képességeitől, körülményeitől függ. Feltételezzük, hogy ez az érték a tanulás folyamán állandó.

Bevezetjük és $A=[a_{ik}]$-val jelöljük $(i, k=1, 2, \ldots, n)$ a tárgyak *relatív disszipáció mátrixának* fogalmát. Az A mátrix a_{ik} eleme azt adja meg, hogy a k-adik tárgy egységnyi intenzitású tanulása milyen mértékben csökkenti a diák terhelhetőségét (elszívja, elemészti, „disszipálja" a diák idejét, energiáját) az i-edik tantárgyban. Az A mátrix a tárgyak „relatív nehézségi fokával" van kapcsolatban. A relatív szó az elnevezésben kétféle értelemben is szerepel. Egyrészt valamely tárgy nehézségi foka a diák adottságaitól is függ; az egyik diáknak az egyik tárgy nehezebb, mint a másik, a másik diáknak fordítva. Másrészt egy rögzített diák esetében az egyes tárgyak nehézségi fokát egymáshoz kell viszonyítani. *A k-adik tárgynak az i-edikre vonatkoztatott relatív nehézségi fokát* a b_i/b_k hányadossal értelmezzük. A pedagógiában régi tapasztalat és általánosan elfogadott nézet az, hogy huzamos ideig egy tárgyra koncentrálni sokkal nehezebb, mint ugyanennyi tanulásra fordított időben több tárgyat váltogatni. Ha például valaki hat órán át fizikát tanult, akkor esetleg aznap már képtelen több fizikát tanulni, de még képes lehet további négy órát filozófiával tölteni. Ennek figyelembevételével a relatív disszipáció mátrix a_{ik} elemét $(i \neq k)$ a következőképpen értelmezzük:

$$a_{ik} = r_{ik}\frac{b_i}{b_k}, \quad i \neq k, \tag{2.1}$$

ahol $0<r_{ik}<1$, $r_{ik}=r_{ki}$ $(i, k=1, 2, \ldots, n; i\neq k)$; az r_{ik} tényező az i-edik és a k-adik tárgy „rokonságát" méri, vagyis azt, hogy mennyire nem üdítő az i-edik tárggyal való foglalkozásról a k-adikkal való foglalkozásra áttérni. Mivel a k-adik tárgy egységnyi intenzitással való tanulása a további terhelhetőséget ebben a tárgyban eggyel csökkenti, ezért

$$a_{kk} = 1, \quad (k = 1, 2, \ldots, n). \tag{2.2}$$

A relatív disszipáció mátrixa tehát olyan pozitív elemű mátrix, melynek főátlójában egyesek állnak.

3. A szabadon tanuló diák modellje

A diákot szabadon tanulónak nevezzük, ha a tanulás folyamata során nem hat rá külső kényszer (feleltetés, dolgozatírás, kötelező házi feladatok elkészítése stb.). Nem számítjuk a külső kényszerek közé azt, hogy ha viszonylag hosszú tanulási idő után a tanult tárgyakból vizsgát kell tennie.

A szabadon tanuló diák $x(t)$ tudásmennyisége mint az idő függvénye az előző pontban ismertetett feltételek mellett, az ott bevezetett jelölésekkel a következő differenciálegyenlet rendszert elégíti ki:

$$\ddot{x} = b - A\dot{x}, \tag{3.1}$$

ahol $b=\mathrm{col}\,(b_1, \ldots, b_n)$ a diák terhelhetőség vektora $(b_i>0, i=1, 2, \ldots, n; n\geqq 2)$, $A=[a_{ik}]$ pedig a relatív disszipáció mátrixa, melynek elemeit a (2.1) és (2.2) formulákkal értelmeztük. Az általunk javasolt (3.1) differenciálegyenlet rendszerben maga az x tudásmennyiség nem szerepel, csupán annak első- és másodrendű derivált függvénye. Ha tehát a tanulás intenzitására bevezetjük az $y=\dot{x}$ jelölést, az intenzitásra az

$$\dot{y}=b-Ay \tag{3.2}$$

elsőrendű állandó együtthatós inhomogén lineáris differenciálegyenlet rendszert nyerjük. Ennek i-edik egyenlete:

$$\dot{y}_i = b_i - \sum_{k=1}^{n} a_{ik} y_k, \quad (i = 1, 2, \ldots, n). \tag{3.3}$$

Ha a tanulás intenzitása az i-edik tárgy kivételével az összes tárgyban zérus, vagyis $y_k = 0$, $k \neq i$, akkor a tudás gyorsulása az i-edik tárgyban (2.2) figyelembevételével

$$\dot{y}_i = b_i - y_i. \tag{3.4}$$

Ezek szerint, ha a diák egyetlen tárgyat, az i-ediket tanulja, akkor a tanulás y_i intenzitása a b_i értékig növelhető. Ha y_i tovább növekedne, deriváltja $\dot{y}_i$ a (3.4) egyenlet szerint negatívvá válna, tehát y_i-nek csökkennie kellene. Ha a diák a többi tárgyat is tanulja, vagyis $y_k > 0$, $(k = 1, 2, \ldots, n)$, akkor tekintve, hogy az A mátrix pozitív elemű, a (3.4) egyenlet jobb oldalából (3.3) szerint további pozitív tagok vonódnak le, tehát $\dot{y}_i$ már valamilyen b_i-nél kisebb y_i érték mellett zérussá válik, és így az i-edik tárgy tanulásának intenzitása nem érheti el a maximális b_i értéket.

Rátérünk a (3.2) egyenlet megoldásainak vizsgálatára. A rendszer durvasága és bizonyos paraméterek, így pl. az r_{ik} mennyiségek erősen szubjektív jellege miatt bizonyos elfajult (strukturálisan nem stabilis) eseteket eleve kizárhatunk. Így feltételezzük, hogy A reguláris mátrix és sajátértékei mind egyszeresek. Jelöljük A sajátértékeit λ_k-val $(k = 1, 2, \ldots, n;\ \lambda_i \neq \lambda_k$, ha $i \neq k)$ és a megfelelő sajátvektorokat $s_1, s_2, \ldots, s_n$-nel; ekkor $-A$ sajátértékei: $-\lambda_1, -\lambda_2, \ldots, -\lambda_n$. A (2.1) formulából következik, hogy az A mátrix hasonló az $[r_{ik}]$ szimmetrikus mátrixhoz, tehát összes λ_k sajátértéke valós.[1] A (3.2) inhomogén egyenlet egy megoldása az

$$y = A^{-1}b$$

állandó. Az általános megoldás

$$y(t) = A^{-1}b + \sum_{k=1}^{n} c_k e^{-\lambda_k t} s_k, \tag{3.5}$$

ahol $c_1, \ldots, c_n$ tetszőleges állandók. (3.5) csak akkor ad valamelyest reális képet a tanulás intenzitásának alakulásáról a $[0, \infty)$ időintervallumban, vagy ennek viszonylag hosszú részintervallumain, ha koordinátái a $[0, \infty)$ intervallumban korlátos függvények. Ennek szükséges és elégséges feltétele az, hogy a $-A$ mátrix összes sajátértéke nem-pozitív legyen, vagyis fennálljon a $\lambda_k \geqq 0$, $(k = 1, 2, \ldots, n)$ feltétel. Azonban zérus sajátérték nem lehetséges, mivel A reguláris mátrix. Tehát a $-A$ mátrix összes sajátértéke negatív:

$$\lambda_k > 0, \quad (k = 1, 2, \ldots, n), \tag{3.6}$$

a (3.2) rendszer aszimptotikusan stabilis, és az összes megoldás az $A^{-1}b$ állandó megoldáshoz tart:

$$\lim_{t \to \infty} y(t) = A^{-1}b. \tag{3.7}$$

[1] Ezt a dolgozat eredményeinek ismertetése során BAJCSAY PÁL vette észre, akinek ezúton fejezem ki köszönetemet.

Világosan látnunk kell, hogy a (3.5) megoldásrendszer egyes megoldásainak mincs gyakorlati jelentősége. Ennek ellenkezőjét feltételezni ugyanis azt jelentené, hogy az intenzitás $y(0)$ kezdeti értéke egyedül determinálja az intenzitás alakulását hosszú időn át. Ez természetesen nem így van, hiszen az intenzitás alakulására sok, többek között véletlen jellegű, tényező hat.

A (3.7) limesz-relációnak azonban gyakorlati jelentőséget kell tulajdonítanunk. Ez azt jelenti, hogy a stabilis esetben (a (3.6) feltételek fennállása esetén), ha külső zavaró tényezők nem jelentkeznek, a szabadon tanuló diák vagy kezdettől fogva, vagy egy bizonyos átmeneti idő eltelte után állandó intenzitással tanulja az összes tárgyat. (A diák természetesen egy időpillanatban csak egyetlen tárgyat tud tanulni. Az intenzitás állandóságát az összes tárgyban viszonylag hosszabb időegységre, például egy hétre vonatkoztatva kell érteni.)

Ha az intenzitás alakulását az időben ismerjük, az $x(t)$ tudásmennyiséget egyszerűen integrálás útján határozhatjuk meg. Az általánosság megszorítása nélkül feltételezhetjük, hogy a diák tudásmennyisége a $t=0$ időpontban az összes tárgyban zérus: $x(0)=0$. Ekkor

$$x(t) = \int_0^t y(\tau)\, d\tau.$$

Ha a diák az $A^{-1}b$ állandó intenzitással tanul (és gyakorlatilag minden esetben amikor a (3.7) limes-reláció fennáll), a tudásmennyiség az időben lineárisan nő:

$$x(t) = A^{-1}bt. \tag{3.8}$$

Az előbbi megfontolások azt mutatják, hogy az intuitív meggondolások alapján konstruált modellnek csak akkor tulajdoníthatunk reális jelentést, ha a $-A$ mátrix stabilis (összes sajátértékének valós része negatív), és az $A^{-1}b$ vektor pozitív elemű.

4. Speciális esetek

Vizsgáljuk először az $n=2$ dimenziós esetet. A (3.2) differenciálegyenlet rendszer ekkor az

$$\dot{y}_1 = b_1 - y_1 - r\frac{b_1}{b_2} y_2$$

$$\dot{y}_2 = b_2 - r\frac{b_2}{b_1} y_1 - y_2 \tag{4.1}$$

alakot ölti, ahol az $r=r_{12}=r_{21}$ jelölést vezettük be $(b_1, b_2>0,\ 0<r<1)$. A $-A$ együttható mátrix karakterisztikus polinomja

$$\det(A+\lambda E) = \lambda^2+2\lambda+(1-r^2),$$

ahol $1-r^2>0$. Ez, mint ismeretes (lásd [2]), azt jelenti, hogy (4.1) aszimptotikusan stabilis rendszer. Egyszerű számolással adódik, hogy az aszimptotikusan stabilis állandó megoldás

$$y = A^{-1}b = \operatorname{col}\left(\frac{1}{1+r} b_1,\ \frac{1}{1+r} b_2\right).$$

Az eredményből látható, hogy a két tárgyat szabadon tanuló diák a b_1, ill. b_2 (maximális) terhelhetőség felénél nagyobb állandó intenzitással képes haladni.

Másodszor az $n=3$ dimenziós esetet vizsgáljuk. A (3.2) rendszerben szereplő mátrix most:

$$A = \begin{vmatrix} 1 & r_{12}\dfrac{b_1}{b_2} & r_{13}\dfrac{b_1}{b_3} \\ r_{12}\dfrac{b_2}{b_1} & 1 & r_{23}\dfrac{b_2}{b_3} \\ r_{13}\dfrac{b_3}{b_1} & r_{23}\dfrac{b_3}{b_2} & 1 \end{vmatrix}.$$

A $-A$ együttható mátrix karakterisztikus polinomja:

$$\det(A+\lambda E) = \lambda^3 + \operatorname{Sp} A\lambda^2 + B\lambda + \det A,$$

ahol $\operatorname{Sp} A = 3$; B az A mátrix másodrendű sarokaldeterminánsainak összege

$$B = 3-(r_{12}^2+r_{23}^2+r_{13}^2);$$

$$\det A = 1+2r_{12}r_{23}r_{13}-(r_{12}^2+r_{23}^2+r_{13}^2).$$

A stabilitás szükséges és elégséges feltétele, hogy teljesüljön a $B>0$, $\det A>0$ és a $B \operatorname{Sp} A > \det A$ egyenlőtlenség (lásd [2]). Mivel $0<r_{ik}<1$ fennáll, ezért az első és a harmadik feltétel automatikusan teljesül. A (3.2) rendszer tehát az $n=3$ esetben akkor és csak akkor aszimptotikusan stabilis, ha

$$\det A = 1+2r_{12}r_{23}r_{13}-(r_{12}^2+r_{23}^2+r_{13}^2) > 0. \tag{4.2}$$

Egyszerű számolással adódik, hogy az állandó megoldás most az

$$y = A^{-1}b = \frac{1}{\det A}\begin{bmatrix} b_1(1-r_{23}^2+r_{13}r_{23}+r_{12}r_{23}-r_{12}-r_{13}) \\ b_2(1-r_{13}^2+r_{23}r_{13}+r_{12}r_{13}-r_{12}-r_{23}) \\ b_3(1-r_{12}^2+r_{23}r_{12}+r_{13}r_{12}-r_{13}-r_{23}) \end{bmatrix}$$

oszlopvektor.

Abban a speciális esetben, amikor bármely két tárgy „rokonsági foka" ugyanaz, vagyis $r_{12}=r_{23}=r_{13}=r$, a (4.2) feltétel az

$$2r^3-3r^2+1 = (r-1)^2(2r+1) > 0$$

alakot veszi fel. Nyilvánvaló, hogy az utóbbi egyenlőtlenség minden $0<r<1$ értékre fennáll. Ebben az esetben az aszimptotikusan stabilis állandó megoldás:

$$y = A^{-1}b = \frac{1}{2r+1}\operatorname{col}(b_1, b_2, b_3).$$

Harmadszor tetszőleges 2-nél nagyobb n-re azt az esetet vizsgáljuk, amikor bármely két tárgy „rokonsági foka" ugyanaz, vagyis

$$r_{ik} = r, \quad (i, k = 1, 2, \ldots, n;\ i \neq k). \tag{4.3}$$

Ekkor a (3.2) rendszerben szereplő mátrix:

$$A = \begin{bmatrix} 1 & r\frac{b_1}{b_2} & r\frac{b_1}{b_3} & \ldots & r\frac{b_1}{b_n} \\ r\frac{b_2}{b_1} & 1 & r\frac{b_2}{b_3} & \ldots & r\frac{b_2}{b_n} \\ r\frac{b_3}{b_1} & r\frac{b_3}{b_2} & 1 & \ldots & r\frac{b_3}{b_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r\frac{b_n}{b_1} & r\frac{b_n}{b_2} & r\frac{b_n}{b_3} & \ldots & 1 \end{bmatrix}.$$

Az A mátrix karakterisztikus polinomja (1 helyébe b_i/b_i-t írva)

$$\det(A - \lambda E) = \begin{vmatrix} \frac{b_1}{b_1} - \lambda & r\frac{b_1}{b_2} & \ldots & r\frac{b_1}{b_n} \\ r\frac{b_2}{b_1} & \frac{b_2}{b_2} - \lambda & \ldots & r\frac{b_2}{b_n} \\ \vdots & \vdots & \ddots & \vdots \\ r\frac{b_n}{b_1} & r\frac{b_n}{b_2} & \ldots & \frac{b_n}{b_n} - \lambda \end{vmatrix} =$$

$$= \frac{b_1 b_2 \ldots b_n}{b_1 b_2 \ldots b_n} \begin{vmatrix} 1-\lambda & r & r & r \\ r & 1-\lambda & r & r \\ \vdots & \vdots & \vdots & \vdots \\ r & r & r & 1-\lambda \end{vmatrix},$$

amit úgy kaptunk, hogy az előző determináns i-edik sorából b_i-t, i-edik oszlopából $\frac{1}{b_i}$-t emeltük ki $(i=1, 2, \ldots, n)$. A sajátértékek (lásd [1]) $\lambda_1 = 1-r$, ami $(n-1)$-szeres, és $\lambda_2 = 1+(n-1)r$. Tehát a $-A$ mátrix összes sajátértéke negatív, és így a rendszer ebben az esetben aszimptotikusan stabilis. Miután mátrix sajátértékei a mátrix elemeinek folytonos függvényei, a (3.2) rendszer akkor is aszimptotikusan stabilis marad, ha az r_{ik} „rokonsági fokok" nem egyenlők, de csak kevéssel térnek el egymástól. Ugyanakkor könnyű példát mutatni arra, hogy ha igen nagy (1-hez közeli) és igen kicsi (0-hoz közeli) „rokonsági fokok" is fellépnek, a rendszer stabilitása megszűnhet. Az egyenlő „rokonsági fokok" feltételezése egyébként láthatólag ellentmond annak a feltételünknek, hogy A összes sajátértéke egyszeres. Ha „kevéssel eltérő rokonsági fokokról" beszélünk, akkor ez az ellentmondás megszüntethető.

5. Tanulás periodikus külső kényszer hatása alatt

Azt modjuk, hogy a tanulás periodikus külső kényszer hatása alatt folyik, ha a diák előrehaladását a tanulmányi idő alatt rendszeresen, szabályos időközönként ellenőrzik, és az ellenőrzés pozitív, illetve negatív eredménye a diák számára előnyös, illetve hátrányos következményekkel jár. Az ellenőrzés lehet feleltetés, zárthelyi dolgozat íratása, házi feladat kidolgoztatása stb. Az előnyös, ill. hátrányos következmények erkölcsiek és anyagiak lehetnek. A közelgő ellenőrzés tudata a normális diákra pszichológiai kényszerként hat, mely a tanulás intenzitásának növelésére serkent. Ugyanakkor általános pedagógiai tapasztalatnak tekinthető az, hogy az ellenőrzés lezajlása után a leglelkiismeretesebb diák is hajlamos a tanulás intenzitásának csökkentésére. A periodikusan ismétlődő ellenőrzést tehát mint az intenzitás változását periodikusan befolyásoló tényezőt kell figyelembe vennünk. Ezt a szabadon tanuló diák modelljéből kiindulva úgy tehetjük meg, hogy a tudás gyorsulását, vagyis az intenzitás időegység alatti megváltozását megadó (3.1), ill. (3.2) egyenlet jobb oldalához egy mind pozitív, mind pedig negatív értékeket felvevő periodikus tagot adunk hozzá.

Tételezzük fel, hogy az összes tárgyban ugyanolyan $\tau>0$ periódussal történik az ellenőrzés, és jelöljük továbbra is $y(t)$-vel a tanulás intenzitását mint az idő függvényét. Ekkor az intenzitás az

$$\dot{y} = b - Ay + f(t) \tag{5.1}$$

differenciálegyenlet rendszert elégíti ki, ahol b és A jelentése ugyanaz, mint a (3.2) egyenletben, $f(t)$ pedig periodikus függvény τ periódussal: $f(t+\tau)\equiv f(t)$.

Miután modellünk amúgy is durva, abból a célból, hogy könnyen kezelhető legyen, az $f(t)$ függvény koordinátáit τ periódusú szinuszfüggvényeknek vesszük fel. Az $\omega=\frac{2\pi}{\tau}$ jelöléssel $f(t)$ i-edik koordinátája:

$$f_i(t) = m_i \sin(\omega t+\delta_i), \quad (i = 1, 2, \ldots, n),$$

ahol feltételezzük, hogy $m_i\geqq 0$. Az m_i, ill. a δ_i számot az *i-edik tárgyban gyakorolt kényszer amplitúdójának,* illetve *kezdőfázisának* nevezzük. Az m_i amplitúdó nagysága függ az i-edik tárgyban gyakorolt ellenőrzés kimenetelének előnyös, illetve hátrányos következményeitől és a diák lelki alkatától. Értéke ezek szerint meglehetősen szubjektív. Annyi bizonyos, hogy a diák b_i terhelhetőségénél (lényegesen?) kisebb. A δ_i kezdőfázisok ($i=1, 2, \ldots, n$) adják meg azt, hogy az ellenőrzési csúcsok a különböző tárgyakban mennyire vannak egymáshoz képest elcsúsztatva az időben.

Az előbbiek figyelembevételével az (5.1) rendszer a következő alakot ölti:

$$\dot{y}_i = -\sum_{k=1}^{n} a_{ik} y_k + b_i + m_i \sin(\omega t+\delta_i), \quad (i = 1, 2, \ldots, n). \tag{5.2}$$

Tételezzük fel, hogy a $-A=[-a_{ik}]$ mátrix stabilis, vagyis összes sajátértéke negatív. Bebizonyítjuk, hogy ekkor az (5.2) rendszernek van τ periódusú harmonikus megoldása, mely a

$$\varphi(t) = A^{-1}b + \psi(t) \tag{5.3}$$

alakban írható, ahol $\psi(t)$ i-edik koordinátája

$$\psi_i(t) = c_i \sin(\omega t + \gamma_i), \quad (i = 1, 2, \ldots, n).$$

Az (5.3) függvényt behelyettesítjük (5.2)-be és megnézzük, milyen feltételt kapunk az egyelőre határozatlan $c = \mathrm{col}(c_1, c_2, \ldots, c_n)$ vektorra és a $\gamma_1, \gamma_2, \ldots, \gamma_n$ kezdőfázisokra:

$$\dot{\psi}(t) \equiv -A(A^{-1}b + \psi(t)) + b + f(t),$$

vagyis

$$\dot{\psi}(t) \equiv -A\psi(t) + f(t).$$

Az utóbbi azonosságot koordinátákra átírva:

$$\omega c_i \cos(\omega t + \gamma_i) \equiv -\sum_{k=1}^{n} a_{ik} c_k \sin(\omega t + \gamma_k) + m_i \sin(\omega t + \delta_i),$$

ahonnan az

$$\omega c_i \cos\gamma_i \cos\omega t - \omega c_i \sin\gamma_i \sin\omega t \equiv \left(m_i \sin\delta_i - \sum_{k=1}^{n} a_{ik} c_k \sin\gamma_k\right) \cos\omega t +$$

$$+ \left(m_i \cos\delta_i - \sum_{k=1}^{n} a_{ik} c_k \cos\gamma_k\right) \sin\omega t, \quad (i = 1, 2, \ldots, n)$$

azonosságot kapjuk. Az utóbbi azonosságból következik, hogy

$$\omega c_i \cos\gamma_i = m_i \sin\delta_i - \sum_{k=1}^{n} a_{ik} c_k \sin\gamma_k,$$

$$-\omega c_i \sin\gamma_i = m_i \cos\delta_i - \sum_{k=1}^{n} a_{ik} c_k \cos\gamma_k, \quad (i = 1, 2, \ldots, n).$$

Vezessük be a következő jelöléseket:

$$c^{(k)} = \mathrm{col}(c_1 \cos\gamma_1,\ c_2 \cos\gamma_2, \ldots, c_n \cos\gamma_n),$$

$$c^{(s)} = \mathrm{col}(c_1 \sin\gamma_1,\ c_2 \sin\gamma_2, \ldots, c_n \sin\gamma_n),$$

$$m^{(k)} = \mathrm{col}(m_1 \cos\delta_1,\ m_2 \cos\delta_2, \ldots, m_n \cos\delta_n),$$

$$m^{(s)} = \mathrm{col}(m_1 \sin\delta_1,\ m_2 \sin\delta_2, \ldots, m_n \sin\delta_n).$$

Ezekkel a jelölésekkel a legutóbbi feltételi egyenletek az

$$\omega c^{(k)} = m^{(s)} - Ac^{(s)},$$

$$-\omega c^{(s)} = m^{(k)} - Ac^{(k)}$$

alakot veszik fel. Szorozzuk meg a második egyenletet $-i$-vel (i: a képzetes egység) és adjuk hozzá az elsőhöz:

$$\omega(c^{(k)} + ic^{(s)}) = m^{(s)} - im^{(k)} + A(ic^{(k)} - c^{(s)}) = iA(c^{(k)} + ic^{(s)}) - i(m^{(k)} + im^{(s)}),$$

vagy rendezve:

$$(A + i\omega E)(c^{(k)} + ic^{(s)}) = m^{(k)} + im^{(s)}.$$

Miután feltevésünk szerint $i\omega$ nem sajátértéke a $-A$ mátrixnak, ezért az $A+i\omega E$ mátrix reguláris, és így

$$c^{(k)}+ic^{(s)} = (A+i\omega E)^{-1}(m^{(k)}+im^{(s)}). \tag{5.4}$$

A $c^{(k)}$ és $c^{(s)}$ vektorok ismeretében a c_i amplitúdók és a γ_i kezdőfázisok meghatározhatók, sőt még azt a további megszorítást is érvényesíthetjük, hogy fennálljanak a $c_i \geqq 0$ egyenlőtlenségek $(i=1, 2, \ldots, n)$.

Az (5.2) rendszer általános megoldása

$$y(t) = y_H(t)+A^{-1}b+\psi(t),$$

ahol $y_H(t)$ a megfelelő homogén rendszer általános megoldása. Mivel feltevéseink szerint az (5.2) rendszer aszimptotikusan stabilis, $t\to\infty$ esetén minden megoldása az (5.3) megoldáshoz tart, vagyis

$$\lim_{t\to\infty}(y(t)-A^{-1}b-\psi(t)) = 0.$$

Ezek szerint szinuszos külső kényszer hatása alatt a tanulás intenzitását gyakorlatilag az (5.3) függvény adja meg, vagyis az intenzitás szinuszosan ingadozik az $A^{-1}b$ állandó körül.

Az előbbiek alkalmazásaként foglalkozzunk a következő speciális esettel. Legyen a tanult tárgyak száma $n=2$, és tételezzük fel, hogy az első tárgyban a tanulás szabadon, a másodikban pedig szinuszos külső kényszer hatása alatt folyik. Azt a kérdést vizsgáljuk, hogyan befolyásolja a második tárgyban alkalmazott külső kényszer az első tárgy tanulását.

Az intenzitás most az

$$\dot{y}_1 = b_1-y_1-r\frac{b_1}{b_2}y_2$$

$$\dot{y}_2 = b_2-r\frac{b_2}{b_1}y_1-y_2+m\sin(\omega t+\delta)$$

differenciálegyenlet rendszert elégíti ki, ahol mivel $m_1=0$, az $m_2=m$, $\delta_2=\delta$ jelöléseket használtuk $(m>0)$. Az (5.4)-nek megfelelő egyenlet:

$$\begin{bmatrix} c_1e^{i\gamma_1} \\ c_2e^{i\gamma_2} \end{bmatrix} = (A+i\omega E)^{-1}\begin{bmatrix} 0 \\ me^{i\delta} \end{bmatrix},$$

ahol

$$(A+i\omega E)^{-1} = \frac{1}{1-r^2-\omega^2+i2\omega}\begin{bmatrix} 1+i\omega & -r\dfrac{b_1}{b_2} \\ -r\dfrac{b_2}{b_1} & 1+i\omega \end{bmatrix}.$$

Így

$$c_1e^{i\gamma_1} = \frac{-\dfrac{b_1}{b_2}mre^{i\delta}}{1-r^2-\omega^2+i2\omega},$$

ahonnan

$$c_1 = \frac{rm}{(\omega^4 + 2(1+r^2)\omega^2 + (1-r^2)^2)^{1/2}} \frac{b_1}{b_2}.$$

A pozitív c_1 szám annak a rezgésnek az amplitúdója, amellyel az első tárgy tanulásának intenzitása rezeg a 4. pont első példájában kiszámított $\frac{1}{1+r} b_1$ állandó érték körül. Ezzel a c_1-gyel mérhetjük a második tárgyban gyakorolt periodikus kényszer zavaró hatását az első tárgyra. Ha megvizsgáljuk a c_1-re kapott kifejezést, a következőket látjuk. c_1 egyenesen arányos a második tárgynak az elsőre vonatkozó relatív nehézségi fokával és a második tárgyban gyakorolt kényszer erősségével. Továbbá c_1 csökken, ha ω vagyis a második tárgyban az ellenőrzések gyakorisága nő. „Ha nagyon gyakran van ellenőrzés, akkor az olyan, mintha sohasem lenne."

6. Lehetőségek és problémák

Az előző pontokban megalkotott és tárgyalt modell valószínűleg a szimultán tanulás folyamatának lehető legegyszerűbb modellje. Alkalmas arra, hogy a tanulás dinamikájának néhány alapvető, kvalitatív jellemzőjét leírja, de finomabb és kvantitatív jellegű vizsgálatoknál minden bizonnyal hamarosan csődöt mond.

A legfőbb kifogás, amit a modellel szemben fel lehet hozni az, hogy ez a modell determinisztikus, holott magát a jelenséget nyilvánvalóan sok véletlen tényező is befolyásolja. Így például a diák b terhelhetősége nyilván változik az időben, mégpedig sztochasztikus módon. A diák lelkiállapota, körülményei előre ki nem számítható módon változhatnak, így a b terhelhetőséget valójában sztochasztikus folyamatnak kell tekintenünk. Ily módon a szimultán tanulás modelljéül egy sztochasztikus differenciálegyenlet rendszert kellene választani. A tanulás intenzitása, mint ennek a sztochasztikus differenciálegyenlet rendszernek a megoldása, maga is sztochasztikus folyamat, vagyis minden rögzített időpontban egy valószínűségi változó. Érdemes lenne kísérletet tenni ilyen sztochasztikus modell megalkotására. Bizonyos egyszerűsítő feltételek fennállása esetén az ebben a dolgozatban vázolt modell valószínűleg alkalmas a várható értékek közötti összefüggések leírására.

A másik irány, amelyben az itt vázolt modellt feltétlenül érdemes általánosítani, gyakorlati megfontolásokkal kapcsolatos. Az esetek nagy részében a tanulás folyamata meghatározott ideig tart és meghatározott tudásmennyiség elérésére irányul (melyről pl. vizsgán számot kell adni). Feltételezhetjük, hogy a tanulás kezdete a $t=0$ időpont, amikor is a tudásmennyiség zérus: $x(0)=0$; a tanulás vége a $t=T>0$ időpont, amikorra is a tudásmennyiségnek el kell érnie az $x(T)=x_T$ értéket.

Induljunk ki a szabadon tanuló diák modelljéből és tételezzük fel, hogy a tudásmennyiség a (3.8) függvény szerint változik. Ha az $A^{-1}b\,T$ vektor nagyobb az x_T vektornál (ezen azt értjük, hogy minden koordinátája nagyobb, mint x_T megfelelő koordinátája), akkor annyit mondhatunk, hogy a diák a megadott időre meg tudja tanulni az anyagot akkor is, ha nem teljes erőbedobással tanul. Ekkor tehát a diák csökkentheti terhelhetőségi vektorát.

Világos, hogy ennél a problémánál a szimultán tanulás folyamatát egy lineáris szabályozási rendszer segítségével érdemes modellezni. Megtartva a $b_1, b_2, \ldots, b_n$

terhelhetőségeknek és az A állandó mátrixnak a 2. és 3. pontban adott értelmezését, a (3.1) differenciálegyenlet rendszert az

$$\ddot{x} = -A\dot{x} + Bu \tag{6.1}$$

szabályozási rendszerrel helyettesítjük, ahol

$$B = \begin{bmatrix} b_1 & 0 & \dots & 0 \\ 0 & b_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_n \end{bmatrix},$$

és $u = \mathrm{col}\,(u_1, u_2, \dots, u_n)$ a $[0, T]$ intervallumon értelmezett szakaszonként folytonos, a $0 \leqq u_i(t) \leqq 1$, $t \in [0, T]$, $(i = 1, 2, \dots, n)$ egyenlőtlenséget kielégítő függvények osztályából választható.

A (6.1) szabályozási rendszerrel kapcsolatban többféle optimalizálási feladat is felvethető. Így elsősorban az időoptimum feladat, hogy ti. mekkora az a minimális T idő, mely alatt az előírt x_T tudásmennyiséget el lehet érni. A (6.1)-gyel adott modell tárgyalására egy későbbi dolgozatban visszatérünk.

Végezetül felhívjuk a figyelmet arra, hogy ebben a dolgozatban bizonyos általánosan elfogadottnak vélt hipotézisek alapján, tisztán elméleti úton konstruáltuk meg a modellt és ebből elméleti úton vontunk le bizonyos következtetéseket. Megállapításainkat semmiféle mérések, vagy kísérletek nem támasztják alá. Ha a szimultán tanulás dinamikája érdekes kérdés, akkor kísérleti adatok gyűjtésére szükség van. Ahhoz azonban, hogy értelmes kísérleteket folytassunk, értékelhető mérési adatokra tegyünk szert, bizonyos elméleti, kiindulási alapra van szükségünk. „Semmi sem gyakorlatiasabb, mint egy jó elmélet." Itt csupán kiindulási alapot kíséreltünk meg adni a mérésekhez és a további elméleti kutatáshoz.

IRODALOM

[1] PARODI, M., *La localisation des valeurs caracteristiques des matrices et ses applications* (Gauthier-Villars, Paris, 1959).
[2] PONTRJAGIN, L. SZ., *Közönséges differenciálegyenletek*, (Akadémiai Kiadó, Budapest, 1972).
[3] SZEKERES, I., „A tanulás folyamatának függvénytani leírása", *Matematikai Lapok* **23** (1972) 319—336.

(Beérkezett: 1976. március 29.)

FARKAS MIKLÓS
BME GÉPÉSZMÉRNÖKI KAR MATEMATIKA TANSZÉK
1521 BUDAPEST XI., STOCZEK U., H ÉPÜLET, IV. EM.

DYNAMIC THEORY OF SIMULTANEOUS LEARNING

M. FARKAS

A deterministic dynamical model is constructed to describe the process of learning in time. It is assumed that the student is learning more than one subject during a long time interval. The model is linear and structurally stable. Thus it yields relatively easily general qualitative conclusions such as the following ones. The intensity of learning is higher if several subjects are studied as compared to the case when only a single one. The intensity is expressed quantitatively. It is constant if learning is done "freely" and it is a periodic function of time if it goes on under regular supervision. The higher is the frequency of regular supervision in one subject the less does it affect the learning of another one.

Alkalmazott Matematikai Lapok **2** *(1976) 115—122*

MEGJEGYZÉSEK ALGEBRAI EGYENLETEK KÖZELÍTŐ MEGOLDÁSÁHOZ

GALÁNTAI AURÉL
Budapest

A dolgozatban az [5], [6] tanulmányban leírt algoritmust hasonlítjuk össze a *Lehmer—Schur eljárás*sal, és egyúttal ennek egy új jellemzését is megadjuk.

1. A vizsgált algoritmusok leírása

Először az [5], [6] dolgozat eljárását ismertetjük. A módszer a *Graeffe-módszer* továbbfejlesztésének is tekinthető. A komplex

$$p_0(z) = \sum_{j=0}^{n} a_{j0} z^j = 0 \quad (a_{j0} \in \mathbf{C},\ a_{00} a_{n0} \neq 0) \tag{1.1}$$

polinom esetén az eljárás általános paraméterezéssel a következő alakban írható.

Jelölje a j-edik *Graeffe-transzformált*at

$$p_j(z) = p_{j-1}(\sqrt{z})\, p_{j-1}(-\sqrt{z}) = \sum_{k=0}^{n} a_{kj} z^k \quad (j = 1, 2, \ldots); \tag{1.2}$$

és legyen rögzített $m_0 \geqq 1$ egész mellett

$$M[p_0(z), m_0] = \left[\max_{k=1,\ldots,n} \left| \frac{\sigma_k}{n} \right|^{\frac{1}{k 2^{m_0}}} \right]^{-1}, \tag{1.3}$$

ahol

$$\sigma_k = k a_{km_0} - \sum_{j=1}^{k-1} a_{jm_0} \sigma_{k-j} / a_{0m_0} \quad (k = 1, \ldots, n). \tag{1.4}$$

Legyen továbbá α_{m_0}, l, m_0 a

$$0{,}5 < \alpha_{m_0} < 5^{-\frac{1}{2^{m_0}}}, \qquad l > \pi \left[\arccos \frac{2{,}5 + \alpha_{m_0}}{2 + 2\alpha_{m_0}} \right]^{-1} - 1, \quad m_0 \geqq 2$$

egyenlőtlenségeknek eleget tevő érték. Ekkor az

$$M^{(0)} = M[p_0(z), m_0], \quad S^{(0)} = 0 \tag{1.5}$$

jelölésekkel az eljárás d-edik lépése ($d = 0, 1, \ldots$) a következő:

1. Algoritmus:

(i) Legyen

$$S_j^{(d+1)} = S^{(d)} + 0{,}5(1+\alpha_{m_0}) M^{(d)} \exp\left(j\frac{2\pi i}{l+1}\right),$$

ahol $j=0, 1, \ldots, l$ és $i=\sqrt{-1}$.

(ii) Ha létezik olyan j index, amelyre $p_0(S_j^{(d+1)})=0$, akkor gyököt kaptunk, és az eljárást befejezzük.

(iii) Ellenkező esetben számítsuk ki az

$$M_j^{(d+1)} = M[p_0(z+S_j^{(d+1)}),\ m_0] \quad (j=0, 1, \ldots, l)$$

mennyiségeket, és legyen

$$M^{(d+1)} = \min_j M_j^{(d+1)} = M_{j(d)}^{(d+1)},\ S^{(d+1)} = S_{j(d)}^{(d+1)}.$$

Belátható, hogy $S^{(d)}$ lineáris sebességgel konvergál $p_0(z)$ valamelyik gyökéhez. Az eljárásról kimutatható továbbá ([5], [6]), hogy adott relatív hiba eléréséhez szükséges lépésszám nem függ a konkrét polinomtól, csak a polinom fokszámától.

A [6] dolgozat utolsó pontjához kapcsolódva dolgozatunk célja az 1. algoritmus összehasonlítása a számítástechnikai gyakorlatban jól bevált *Lehmer—Schur módszer*rel ([2—4]). Az eljárás a következőképpen foglalható össze.

Legyen

$$T[p_0(z)] = \sum_{j=0}^{n-1} (\bar{a}_{00}a_{j0} - a_{n0}\bar{a}_{n-j,0})z^j \tag{1.6}$$

és

$$T^j[p_0(z)] = T\{T^{j-1}[p_0(z)]\} \quad (j=2, \ldots). \tag{1.7}$$

Képezzük a $c_j=T^j[p_0(0)]$ $(j=1, \ldots, k)$ számokat, ahol

$$k = \min\{m\in\mathbf{N}|\ c_m = 0\}. \tag{1.8}$$

A $\{c_j\}_{j=1}^k$ számok segítségével a következő függvényt definiáljuk:

$$N[p_0(z)] = \begin{cases} 1, & \text{ha van } j\in\{1, \ldots, k-1\}, \text{ amelyre } c_j<0 \\ 0, & \text{ha } c_j>0\ (j=1, \ldots, k-1) \text{ és } \operatorname{gr} T^{j-1}[p_0(z)]=0 \\ -1, & \text{egyébként.} \end{cases}$$

Ismeretes, hogy $N[p_0(z)]=1$ esetén a $p_0(z)$ polinomnak van gyöke a $\{z\in\mathbf{C}||z|<1\}$ nyílt egységkörben, $N[p_0(z)]=0$ esetén pedig nincs. Az $N[p_0(z)]=-1$ esetre még visszatérünk.

Vezessük be az

$$\alpha_j^{(d)} = \begin{cases} 0{,}5\gamma_0^{(d)} R^{(d-1)} & (j=0), \\ 0{,}4\gamma_j^{(d)} R^{(d-1)} & (j=1, \ldots, 8) \end{cases} \tag{1.9}$$

és

$$\beta_j^{(d)} = \begin{cases} z^{(d-1)} & (j=0), \\ z^{(d-1)} + \dfrac{0{,}75R^{(d-1)}}{\cos\dfrac{\pi}{8}} \exp\left(\dfrac{2\pi i(j-1)}{8}\right) & (j=1, \ldots, 8) \end{cases} \tag{1.10}$$

jelöléseket, ahol az $\{R^{(d)}\}$, $\{z^{(d)}\}$ és $\{\gamma_j^{(d)}\}$ sorozatokat a *Lehmer—Schur algoritmus* d-edik lépésében $(d=1, \ldots)$ határozzuk meg. Legyen $\tilde{p}_0(z)=p_0(z)/\psi$ $(\psi>0)$ és

$$z^{(0)} = 0, \quad R^{(0)} = 1+\max_j \left|\frac{a_{j0}}{a_{n0}}\right|. \tag{1.11}$$

2. Algoritmus:

(i) Ha létezik olyan j index, amelyre $p_0(\beta_j^{(d)})=0$, akkor gyököt kaptunk és az eljárást befejezzük.

(ii) Válasszunk olyan $j\in\{0, 1, \ldots, 8\}$ indexet, amelyre

$$N[\tilde{p}_0(\alpha_j^{(d)} z+\beta_j^{(d)})] = 1$$

és legyen

$$z^{(d)} = \beta_j^{(d)}, \quad R^{(d)} = \alpha_j^{(d)}.$$

A $\gamma_j^{(d)}\in[1, 1+\delta]$ $(\delta\leqq 0, 5)$ számokat úgy választjuk meg, hogy teljesüljön $N[p_0(\alpha_j^{(d)} z+\beta_j^{(d)})]\geqq 0$ (1 valószínűséggel $\gamma_j^{(d)}=1$ választható). Az eljárásról kimutatható, hogy $z^{(d)}$ lineáris sebességgel konvergál $p_0(z)$ valamelyik gyökéhez. Használata esetén adott $\varepsilon>0$ pontosságú (abszolút hibájú) közelítés eléréséhez a polinomtól függő lépésszám szükséges.

2. Hibaanalízis

Kimutatjuk az 1. algoritmus numerikus instabilitását a $\{z\in\mathbf{C}\,|\,|z|\geqq 1+\eta\}$ tartományban, ahol $\eta=\eta(\varepsilon)>0$.

Legyenek az (1.1) polinom gyökei $(z_1, \ldots, z_n)$ úgy indexezve, hogy fennálljon

$$|z_1| \geqq |z_2| \geqq \ldots \geqq |z_n|. \tag{2.1}$$

Belátható ([5], [6]), hogy ekkor

$$5^{-\frac{1}{2^{m_0}}} \leqq \frac{|z_n|}{M[p_0(z), m_0]} \leqq 1. \tag{2.2}$$

A numerikus instabilitást először egy konkrét példán mutatjuk ki

PÉLDA. Számítsuk ki $M[p_0(z), m_0]$ értékét a

$$p_0(z) = (z-i)(z+i)(z-2)(z+3)(z-8) = 0 \tag{2.3}$$

és

$$p_0^*(z) = (z-1, 1i)(z+1, 1i)(z-2)(z+3)(z-8) = 0 \tag{2.4}$$

polinomokra. CDC 3300-as számítógépen kapott eredményeink $\varepsilon=10^{-8}$ pontossággal a következők

m_0	$M[p_0(z), m_0]$	$M[p_0^*(z), m_0]$
3	0.989764735	0.896706515
4	0.967111068	0.878788293
5	0.958606057	0.869992549

Látható, hogy a számított értékekre a (2.2) szerinti

$$|z_n| \leqq M[p_0(z), m_0] \tag{2.5}$$

reláció helyett $|z_n| > M[p_0(z), m_0]$ áll fenn, amely a $p_0^*(z)$ polinom esetén 20%-os relatív hibát jelent.

Az instabilitást most egy általános esetben is megmutatjuk.

Tegyük fel, hogy az (1.4) rekurzió σ_k-val jelölt pontos, és σ_k^*-gal jelölt számított megoldására

$$|\sigma_k - \sigma_k^*| < \varepsilon \quad (k = 1, \ldots, n) \tag{2.6}$$

fennáll. Ekkor a $\delta f(a) = f'(a)\delta a + 0((\delta a)^2)$ relációt használva azt kapjuk, hogy

$$\delta M[p_0(z), m_0] = \frac{\varepsilon}{k_0 n 2^{m_0+1}} \left|\frac{\sigma_{k_0}}{n}\right|^{-2-\frac{1}{k_0 2^{m_0}}} + O(\varepsilon^2) \quad (k_0 \in \{1, \ldots, n\}), \tag{2.7}$$

amelynek főtagja elég kicsiny σ_{k_0} ($|\sigma_{k_0}| < n$) esetén tetszőleges nagy lehet.

Az (1.1) polinomra fennálló (2.2) egyenlőtlenség miatt a $|\sigma_{k_0}| < n$ esetben $|z_n| \geqq 5^{-\frac{1}{2m_0}}$.

Végül megmutatjuk, hogy $|z_n| \geqq 1$ esetén $|\sigma_{k_0}| \leqq n$, illetve pontosabban, adott $\varepsilon > 0$ esetén

$$\delta M[p_0(z), m_0] = O(|z_n|^\tau) \tag{2.8}$$

ahol $\tau \geqq 2^{m_0} + 1$.

A (2.2) egyenlőtlenség és (2.7) alapján

$$\left|\frac{\sigma_{k_0}}{n}\right|^{-2-\frac{1}{k_0 2^{m_0}}} \geqq |z_n|^{k_0 2^{m_0+1}+1}, \tag{2.9}$$

amiből a bizonyítandó állítás már következik.

Ha $\varepsilon > 0$ rögzített és $|z_n| \to \infty$, akkor

$$\delta M[p_0(z), m_0] \to \infty, \tag{2.10}$$

amivel a numerikus instabilitást igazoltuk.

Az instabilitást az (1.3) képletbeli reciprokképzés, és a k-adik gyökvonások együttesen hozzák létre.

3. Az algoritmusok korlátos lefuttathatósága

Jelölje az n-edfokú komplex polinomok halmazát $\mathscr{P}_n$, az egész számokét pedig $\mathbf{Z}$.

Az M numerikus módszert (iterációs eljárást), amely a $p_0(z) \in \mathscr{P}_n$ polinom egy gyökét határozza meg, azonosíthatjuk az algoritmusban fellépő $\{b_k\} \subset \mathbf{C}$ számsorozattal. A $\{b_k\}$ sorozat alkalmas $\{b_{k_j}\}$ részsorozatára tehát fennáll

$$z^* = \lim_{j \to \infty} b_{k_j}, \quad p_0(z^*) = 0. \tag{3.1}$$

A $p_0(z)$ polinomtól függő $\{b_k\}$ sorozatot az $\{Mp_0\} = \{b_k\}$ szimbólummal jelöljük.

Ismeretes, hogy a digitális számítógépek az elemi aritmetikai műveleteket csak a véges

$$S[0, K]\cap \mathbf{C}_\delta \tag{3.2}$$

halmazon tudják végrehajtani, ahol $S[0, K]=\{z\in\mathbf{C}||z|\leqq K\}$ és

$$\mathbf{C}_\delta = \{z\in\mathbf{C}|z = k\delta+j\delta i;\ k, j\in\mathbf{Z}\} \quad (\delta > 0). \tag{3.3}$$

Ha a $\{b_k\}$ sorozatban valamely b_{k_0} elemre $|b_{k_0}|>K$, akkor az algoritmus a számítógépen tovább már nem folytatható, mert túlcsordulás áll elő.

Az aritmetikai túlcsordulás vizsgálatához bevezetjük a korlátosan megoldható (számítógépen lefuttatható) problémák

$$\mathscr{P}_M(a, K, K^*) = \{p_0(z)\in\mathscr{P}(a,K^*)|\{Mp_0\}\subset S[0, K],\quad |\{Mp_0\}| = \infty\} \tag{3.4}$$

osztályát, ahol

$$\mathscr{P}(a,K^*) = \{p_0(z)\in\mathscr{P}_n|\ 0 < |z_j| \leqq a \quad (j = 1, \ldots, n),\ \|p_0(z)\| \leqq K^*\} \tag{3.5}$$

és

$$\|p_0(z)\| = \max_j |a_{j0}|. \tag{3.6}$$

Az $|\{Mp_0\}|$ a $\{b_k\}$ sorozat számosságát jelöli. Érvényesek a következő állítások.

3.1. Állítás. Az 1. algoritmushoz tartozó $\mathscr{P}_1(a, K, K^*)$ halmaz tetszőleges $a, K, K^*>0$ esetén üres.

Bizonyítás. Az 1. algoritmus konvergenciája $(S^{(d)}\to z^*, p_0(z^*)=0)$ azt jelenti, hogy a $p_0(z+S^{(d)})$ $(d=0, 1, \ldots)$ polinomok $z_n^{(d)}$ minimális abszolútértékű gyökeinek sorozatára

$$|z_n^{(d)}| \leqq cq^d \quad (c > 0,\ 0 < q < 1) \tag{3.7}$$

teljesül. A (2.2) egyenlőtlenséget felhasználva azt kapjuk, hogy $d\geqq d'$ esetén

$$\frac{n}{5c}\left(\frac{1}{q}\right)^{d2^{m_0}} \leqq \frac{n}{5|z_n^{(d)}|^{2^{m_0}}} \leqq |\sigma_{k(d)}^{(d)}| \tag{3.8}$$

ahol $k(d)\in\{1, \ldots, n\}$ az (1.3) szerinti maximális index és

$$\frac{n}{5c}\left(\frac{1}{q}\right)^{d'2^{m_0}} > 1. \tag{3.9}$$

Minthogy $|\sigma_{k(d)}^{(d)}|=O(w^d)$ nagyságrendben növekszik $\left(w=\left(\frac{1}{q}\right)^{2^{m_0}}\right)$, elég nagy d_0 indexre $|\sigma_{k(d)}^{(d)}|>K$ $(d\geqq d_0)$, amivel állításunkat igazoltuk.

3.2. Állítás. Ha $K\geqq K^*2^{n+1}(1+a^n2^n)^{n+1}+1$, akkor a 2. algoritmus esetén

$$\mathscr{P}_2(a, K, K^*) = \mathscr{P}(a, K^*). \tag{3.10}$$

Bizonyítás. Egyszerű számításokkal igazolható, hogy az algoritmusban előforduló számítási elemekre érvényesek a

$$|p_0(\beta_j^{(d)})| \leqq \begin{cases} \|p_0(z)\| 2^{n+1} & (a < 0{,}5), \\ \|p_0(z)\|(1+2^n a^n)^{n+1} & (a \geqq 0{,}5), \end{cases} \tag{3.11}$$

$$\|T^j[p_0(z)\| \leqq \frac{1}{2}(2\|p_0(z)\|)^{2^j} \quad (j = 1, \ldots, n) \tag{3.12}$$

és

$$\|p_0(\alpha_j^{(d)} z + \beta_j^{(d)})\| \leqq \|p_0(z)\|(2+2^{n+1}a^n)^n \quad (j = 0, 1, \ldots, 8) \tag{3.13}$$

egyenlőtlenségek ($d = 0, 1, \ldots$); ahonnan a

$$\delta = \|p_0(z)\|(2+2^{n+1}a^n)^n$$

jelöléssel

$$\|T^k[p_0(\alpha_j^{(d)} z + \beta_j^{(d)})]\| \leqq \frac{1}{2}(2\delta)^{2^k} \quad (k = 1, \ldots, n). \tag{3.14}$$

Minthogy K nagyobb, mint (3.11) és (3.13) jobb oldala, és alkalmas lenormálásokkal ($\psi > 2\delta$) a $\tilde{\delta} < 0{,}5$ elérhető, állításunk igaz.

A módszerek közti különbséget az okozza, hogy az 1. algoritmus a (2.2) direkt becslést használja, a *Lehmer—Schur eljárás* pedig az $N[p_0(z)]$ karakterisztikus függvényt, amely a $p_0(z) \to p_0(z)/\psi$ $(0 < \psi \leqq K)$ transzformációval szemben invariáns.

Az 1. algoritmuson hasonló módosítás nem segít, mert a (3.8) relációból következik, hogy az a

$$p_0(z) \to p_0(z)/\psi, \quad p_0(z) \to p_0(z/\psi) \quad (0 < \psi < K) \tag{3.15}$$

transzformáltak használata esetén is túlcsordul.

A számítógépeken leggyakrabban a $K \leqq 10^{256}$, $K^* \geqq 1$ korlátok fordulnak elő, az utóbbi stabilitási okok miatt. A (3.11)—(3.14) becslésekkel együtt főleg ez a tény okozza a magasabb fokszámú polinomok ($n \geqq 30$) számítógépes megoldásának nehézségeit.

4. A számítási költségek vizsgálata

Az előző pontban kimutattuk, hogy az 1. algoritmus nem futtatható le korlátosan. Minthogy adott $\varepsilon > 0$ abszolút hibájú közelítő megoldás kiszámítható a korlátos $S[0, \tilde{K}]$ halmazban is, ahol a $\tilde{K}$ sugár a $p_0(z)$ polinomtól, ε-tól és az algoritmustól függ, az algoritmusok további elemzése szükséges.

A j-edik algoritmus költségén $j = 1, 2$ a lépésenként szükséges multiplikatív és additív műveletek K_{mj} és K_{aj} számát értjük.

Feltéve, hogy a k-adik gyökvonások gépideje 3 additív és 3 multiplikatív műveletével azonos (ezzel a gépidőt erősen alábecsültük) az 1. algoritmus költsége

$$K_{m1} = (l+1)(m_0+4)\frac{n^2}{2} + (l+1)(m_0+8)\frac{n}{2} + O(1) \tag{4.1}$$

és

$$K_{a1} = (l+1)(m_0+4)\frac{n^2}{4} + (2l+3)n + O(1). \tag{4.2}$$

A 2. algoritmus költségére pedig teljesül, hogy

$$K_{m2} \leqq 27n^2 - 18n \tag{4.3}$$

és

$$K_{a2} \leqq 9n^2 + 36n. \tag{4.4}$$

Ha a (4.3)—(4.4) költségkorlátokat egy lépés költségének tekintjük, akkor a 2. algoritmus sebessége

$$|z^{(d)} - z^*| \leqq c_2 \left(\frac{2}{5}\right)^d \quad (d = 0, 1, \ldots). \tag{4.5}$$

Az 1. algoritmus sebessége

$$|S^{(d)} - z^*| \leqq c_1 [q(\alpha_{m_0}, m_0, l)]^d \quad (d = 0, 1, \ldots), \tag{4.6}$$

ahol

$$q(\alpha_{m_0}, m_0, l) = \left[1 + 0{,}25(1+\alpha_{m_0})^2 - (1+\alpha_{m_0}) \cos\frac{\pi}{l+1}\right]^{1/2} \alpha_{m_0}^{-1}. \tag{4.7}$$

Ha a $\delta = \dfrac{(m_0+4)(l+1)}{54} > 1$ és $n \geqq n'$ relációk fennállnak, akkor

$$K_{m1} \leqq \delta K_{m2},\ K_{a1} > \delta K_{a2}. \tag{4.8}$$

4.1. ÁLLÍTÁS. Ha $l \geqq l'$, akkor

$$q(\alpha_{m_0}, m_0, l) > \left(\frac{2}{5}\right)^{\delta}. \tag{4.9}$$

Bizonyítás. Elég nagy l' és $l \geqq l'$ esetén

$$q(\alpha_{m_0}, m_0, l)^2 \geqq \frac{1 - \left(\cos\dfrac{\pi}{l+1}\right)^2}{\alpha_{m_0}^2} \geqq \frac{9\alpha_{m_0}^{-2}}{(l+1)^2}$$

és

$$\left(\frac{5}{2}\right)^{\delta} > l+1,$$

amiből állításunk következik.

Az $l \geqq l'$ esetben az 1. algoritmus d lépésének összköltsége árán a *Lehmer—Schur eljárással* $[\delta d]$ lépést tehetünk. A 4.1. állítás miatt

$$c^*[q(\alpha_{m_0}, m_0, l)]^d > \left(\frac{2}{5}\right)^{[\delta d]} \quad (c^* > 0;\ d \geqq d_0), \tag{4.10}$$

ezért a *Lehmer—Schur eljárás* az additív és multiplikatív műveletekre felbontva gyorsabb az 1. algoritmusnál, ha azt is felbontjuk multiplikatív és additív műveletekre.

Az [5], [6] dolgozatokban szereplő paraméterekre a (4.10) reláció szintén teljesül, amit (4.9) alapján könnyen verifikálhatunk.

A [6] dolgozatban hivatkozás történik az ún. végtelen pontosságú egész aritmetikákra ([1]), amelyek, mint ismeretes, elvileg tetszőleges nagyságú számokkal történő műveleteket tesznek lehetővé.

Ismeretes továbbá az is, hogy a multiplikatív műveletek gépideje azonos hosszúságú számok esetén

$$l(x)^{1+\tau} \quad (\tau > 0) \tag{4.11}$$

időegység, ahol x az $l(x)$ egész szám kettes számrendszerbeli hossza ([1]).

A (3.8) egyenlőtlenség miatt az 1. algoritmus legalább $\alpha 2^{m_0-1}$-szer olyan hosszú számokkal dolgozik, mint a 2. algoritmus $(0,9 \leqq \alpha < 1, 0 < \varepsilon \leqq \varepsilon_0)$. A gépidőben kifejezett költségekre tehát fennáll

$$K_{m1}(t) \cong \delta(\alpha 2^{m_0-1})^{1+\tau} K_{m2}(t) \tag{4.12}$$

és (4.10)-ben is δ helyett $\tilde{\delta} = \delta\alpha 2^{m_0-1}$ írható, ami a *Lehmer—Schur algoritmus* relatív sebességét tovább növeli.

Végül köszönetemet fejezem ki TURÁN PÁL professzornak megjegyzéseiért és a dolgozat témájáért.

IRODALOM

[1] COLLINS, G., "Computer algebra of polynomials and rational functions", *Amer. Math. Monthly* **80** (1973) 725—755.
[2] LEHMER, D. H., "A machine method for solving polynomial equations", *JACM* **8** (1961) 151—163.
[3] RALSTON, A., *Bevezetés a numerikus analízisbe*, (Műszaki Könyvkiadó, Budapest, 1969).
[4] SZIDAROVSZKY, F., *Bevezetés a numerikus módszerekbe* (Közgazdasági és Jogi Könyvkiadó, Budapest, 1974).
[5] TURÁN, P., „Algebrai egyenletek közelítő megoldásáról", *MTA III. Osztály Közleményei* **18** (1968) 223—235.
[6] TURÁN, P., "Power sum method and the approximative solution of algebraic equations", *Math. of Comput.* **29** (1975) 311—318.

(Beérkezett: 1976. január 15.)

DR. GALÁNTAI AURÉL
ELTE TTK NUMERIKUS ÉS GÉPI MATEMATIKA TANSZÉK
1088 BUDAPEST VIII., MÚZEUM KRT. 6—8.

REMARKS ON APPROXIMATE SOLUTION OF ALGEBRAIC EQUATIONS

A. GALÁNTAI

This paper compares the algorithm presented in [5], [6] with the *Lehmer—Schur's process*. A new characterization of the *method Lehmer—Schur* is also given.

Alkalmazott Matematikai Lapok **2** *(1976) 123—125*

MEGJEGYZÉS A NEWTON—MOSER TÍPUSÚ ITERÁCIÓKHOZ

SZIDAROVSZKY FERENC
Budapest

J. MOSER [2] dolgozatában az $f(x)=0$ egyismeretlenes egyenlet megoldására az

$$x_{k+1} = x_k - y_k f(x_k), \tag{1}$$

$$y_{k+1} = y_k - y_k[f'(x_k)y_k - 1] \tag{2}$$

iterációs eljárást javasolta, és kimutatta, hogy ha x^* az egyenlet egyszeres gyöke, akkor megfelelő differenciálhatósági feltételek teljesülése esetén az eljárás konvergens, és a konvergencia sebessége $(\sqrt{5}+1)/2$ rendű. Az (1) egyenlet $y_k = 1/f'(x_k)$ esetén a *Newton-módszert* adja, a (2) egyenlet pedig a *Newton-módszer* alkalmazása az $1/y - f'(x_k) = 0$ egyenletre.

O. H. HALD [1] dolgozatában J. MOSER módszerének *Seidel-típusú* változatával, az

$$x_{k+1} = x_k - y_k f(x_k), \tag{3}$$

$$y_{k+1} = y_k - y_k[f'(x_{k+1})y_k - 1] \tag{4}$$

eljárással foglalkozik, és alkalmas feltételek mellett kimutatja a módszer kvadratikus konvergenciáját. Ugyanakkor részletesen foglalkozik eljárásának numerikus tulajdonságaival, számítógépes realizálhatóságával.

Dolgozatunkban az (1), (2) eljárásnál általánosabb

$$\mathbf{x}^{(k+1)} = \mathbf{g}(\mathbf{x}^{(k)}) \tag{5}$$

iterációval foglalkozunk, ahol $\mathbf{x}^{(k)}, \mathbf{g}(\mathbf{x}^{(k)}) \in R^n$. Tegyük fel, hogy $\mathbf{x}^*$ az $\mathbf{x} = \mathbf{g}(\mathbf{x})$ egyenlet egy megoldása, valamint $\mathbf{x}^*$ valamilyen B konvex környezetében $\mathbf{g}$ kétszer korlátosan differenciálható. Jelölje $\mathbf{x}^*, \mathbf{x}^{(k)}, \mathbf{g}$ komponenseit rendre x_i^*, $x_i^{(k)}, g_i$ $(1 \leqq i \leqq n)$. Bebizonyítjuk a következő tételt.

TÉTEL. Ha $i \leqq j$ esetén

$$\frac{\partial g_i(\mathbf{x}^*)}{\partial x_j} = 0, \tag{6}$$

akkor az (5) módszernek megfelelő

$$
\begin{aligned}
x_1^{(k+1)} &= g_1(x_1^{(k)}, x_2^{(k)}, \ldots, x_{n-1}^{(k)}, x_n^{(k)}),\\
x_2^{(k+1)} &= g_2(x_1^{(k+1)}, x_2^{(k)}, \ldots, x_{n-1}^{(k)}, x_n^{(k)}),\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
x_n^{(k+1)} &= g_n(x_1^{(k+1)}, x_2^{(k+1)}, \ldots, x_{n-1}^{(k+1)}, x_n^{(k)})
\end{aligned} \tag{7}
$$

Seidel típusú eljárás kvadratikus konvergenciával rendelkezik.

Bizonyítás. Vezessük be a

$$
\begin{aligned}
\psi_1(\mathbf{x}) &= g_1(\mathbf{x})\\
\psi_2(\mathbf{x}) &= g_2(\psi_1(\mathbf{x}), x_2, \ldots, x_{n-1}, x_n)\\
\psi_3(\mathbf{x}) &= g_3(\psi_1(\mathbf{x}), \psi_2(\mathbf{x}), \ldots, x_{n-1}, x_n)\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
\psi_n(\mathbf{x}) &= g_n(\psi_1(\mathbf{x}), \psi_2(\mathbf{x}), \ldots, \psi_{n-1}(\mathbf{x}), x_n)
\end{aligned} \tag{8}
$$

sorozatot. Nyilvánvaló, hogy (7) az

$$\mathbf{x}^{(k+1)} = \psi(\mathbf{x}^{(k)}) \tag{9}$$

alakba írható, ahol ψ komponenseit $\psi_1, \psi_2, \ldots, \psi_n$ jelöli. Egyszerű számolással látható be, hogy $i \leqq j$ esetén

$$\frac{\partial \psi_i(\mathbf{x}^*)}{\partial x_j} = \frac{\partial g_i(\mathbf{x}^*)}{\partial x_j} + \sum_{l=1}^{i-1} \frac{\partial g_i(\mathbf{x}^*)}{\partial x_l} \cdot \frac{\partial \psi_l(\mathbf{x}^*)}{\partial x_j}, \tag{10}$$

valamint $i > j$ esetén

$$\frac{\partial \psi_i(\mathbf{x}^*)}{\partial x_j} = \sum_{l=1}^{i-1} \frac{\partial g_i(\mathbf{x}^*)}{\partial x_l} \cdot \frac{\partial \psi_l(\mathbf{x}^*)}{\partial x_j}, \tag{11}$$

amelyből indukcióval azonnal adódik, hogy $i, j = 1, 2, \ldots, n$ esetén

$$\frac{\partial \psi_i(\mathbf{x}^*)}{\partial x_j} = 0. \tag{12}$$

Tehát a ψ függvény *Jacobi-mátrixa* az $\mathbf{x}^*$ pontban eltűnik. Jelölje $\mathbf{H}_i(\mathbf{x})$ a ψ_i függvény *Hesse-féle mátrixát*, azaz $\mathbf{H}_i$ k-adik sorának j-edik eleme $\dfrac{\partial^2 \psi_i(\mathbf{x})}{\partial x_k \partial x_j}$. A többváltozós *Taylor-formula* felhasználásával azonnal adódik, hogy $i = 1, 2, \ldots, n$ esetén

$$x_i^{(k+1)} - x_i^* = \psi_i(\mathbf{x}^{(k)}) - \psi_i(\mathbf{x}^*) = (\mathbf{x}^{(k)} - \mathbf{x}^*)^T \mathbf{H}_i(\xi_k)(\mathbf{x}^{(k)} - \mathbf{x}^*), \tag{13}$$

ahol T a transzponálás jele, valamint ξ_k az $\mathbf{x}^*$ és $\mathbf{x}^{(k)}$ pontot összekötő egyenes szakasz egy pontja. A (13) egyenlőségből közvetlenül leolvasható, hogy

$$|x_i^{(k+1)} - x_i^*| \leqq \sup_{\xi_k \in B} \|\mathbf{H}_i(\xi_k)\|_\infty \|\mathbf{x}^{(k)} - \mathbf{x}^*\|_\infty^2, \tag{14}$$

ahol tetszőleges kvadratikus mátrix $\mathbf{A}=(a_{ij})_{i,j=1}^{n}$ és vektor $\mathbf{u}=(u_i)_{i=1}^{n}$ esetén

$$\|\mathbf{A}\|_\infty = \max_{1\leqq i\leqq n} \sum_{j=1}^{n} |a_{ij}| \quad \text{és} \quad \|\mathbf{u}\|_\infty = \max_{1\leqq i\leqq n} |u_i|. \tag{15}$$

Legyen

$$K = \max_{1\leqq i\leqq n} \left\{ \sup_{\xi_k \in B} \|\mathbf{H}_i(\xi_k)\|_\infty \right\}, \tag{16}$$

ekkor (14) alapján

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\|_\infty \leqq K \cdot \|\mathbf{x}^{(k)} - \mathbf{x}^*\|_\infty^2, \tag{17}$$

vagyis a

$$\delta_k = K \cdot \|\mathbf{x}^{(k)} - \mathbf{x}^*\|_\infty \tag{18}$$

mennyiségek kielégítik a

$$\delta_{k+1} \leqq \delta_k^2 \leqq \ldots \leqq \delta_0^{2^{k+1}} \tag{19}$$

egyenlőtlenséget. Tegyük fel, hogy olyan $\mathbf{x}^{(0)}$ kezdeti közelítést választunk, amelyre

$$\|\mathbf{x}^{(0)} - \mathbf{x}^*\| < \frac{1}{K}, \tag{20}$$

valamint a

$$G = \{\mathbf{x} | \mathbf{x} \in R^n, \|\mathbf{x} - \mathbf{x}^*\|_\infty \leqq \|\mathbf{x}^{(0)} - \mathbf{x}^*\|_\infty\} \tag{21}$$

halmaz része B-nek. Ekkor a (7) eljárás konvergens és (19) alapján a konvergencia kvadratikus.

1. Megjegyzés. A (16), (18) és (19) egyenlőtlenség alapján konkrét hibaformulák nyerhetők, a számolás részleteitől eltekintünk.

2. Megjegyzés. A tétel feltételei az (1), (2) eljárás esetében nyilvánvalóan teljesülnek, így a (3), (4) eljárás kvadratikus konvergenciája a fenti tételből közvetlenül adódik.

IRODALOM

[1] HALD, O. H., "On a Newton—Moser Type Method", *Numer. Math.* **23** (1975) 411—426.
[2] MOSER, J., *Stable and Random Motions in Dynamical Systems with Special Emphasis on Celestial Mechanics* (Herman Weyl Lectures, Annales of Mathematics Studies, no. 77. Princeton, New Jersey: Princeton University Press, 1973).
[3] SZIDAROVSZKY, F., *Bevezetés a numerikus módszerekbe.* (Közg. és Jogi Könyvkiadó, Budapest, 1974.)

(Beérkezett: 1975. október 23.)

SZIDAROVSZKY FERENC
ELTE TTK NUMERIKUS ÉS GÉPI MATEMATIKA TANSZÉK
1088 BUDAPEST VIII., MÚZEUM KRT. 6—8.

REMARK ON THE NEWTON—MOSER TYPE ITERATIONS

F. SZIDAROVSZKY

In this paper a resulf of O. H. HALD is generalized. It is proved then if function g is twice differentiable on a neighbourhood of a root x^* of the equation $x=g(x)$ and the second ordered derivatives are bounded, furthemore for $i\leqq j$ equation (6) holds, then the *Seidel-type iteration method* (7) has quadratic convergence. As a special case the quadratic convergence of method (3), (4) is guaranteed.

Alkalmazott Matematikai Lapok **2** *(1976) 127—134*

NUMERIKUS MÓDSZER NEMLINEÁRIS EGYENLETRENDSZEREK MEGOLDÁSÁRA

GERGELY JÓZSEF
Budapest

A cikk egy iterációs eljárást ajánl nemlineáris egyenletrendszer megoldására. Működését összehasonlítja más módszerekkel. Az eljárást alkalmazza nemlineáris peremfeladat megoldására. Elemzi a számítógépes tapasztalatokat.

1. Az iterációs eljárás ismertetése

Tekintsük az

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})) = \mathbf{0}, \quad \mathbf{x} \in R^n \tag{1.1}$$

nemlineáris egyenletrendszert. Megoldására a következő iterációs eljárást javasoljuk. Valamilyen $\mathbf{x}^0 = (x_1^0, x_2^0, \ldots, x_n^0)$ közelítésből kiindulva oldjuk meg először az

$$f_1(x_1^0 + \delta x_1, x_2^0, \ldots, x_n^0) = 0$$

egyenletet δx_1-re. Legyen a megoldása δx_1^0 és $x_1^1 = x_1^0 + \delta x_1^0$. Ezután oldjuk meg az

$$f_2(x_1^1 + c_1^1 \delta x_2, x_2^0 + \delta x_2, x_3^0, \ldots, x_n^0) = 0$$

egyenletet δx_2-re. A c_1^1 együtthatót a

$$df_1 = \frac{\partial f_1}{\partial x_1} \delta x_1 + \frac{\partial f_1}{\partial x_2} \delta x_2 \equiv 0$$

összefüggésből számoljuk, ahonnan

$$\delta x_1 = -\frac{\dfrac{\partial f_1}{\partial x_2}}{\dfrac{\partial f_1}{\partial x_1}} \delta x_2 = c_1^1 \delta x_2 \quad \text{és} \quad c_1^1 = -\frac{\dfrac{\partial f_1}{\partial x_2}}{\dfrac{\partial f_1}{\partial x_1}}.$$

$\left(\text{A } \dfrac{\partial f_1}{\partial x_2} \text{ és } \dfrac{\partial f_1}{\partial x_1} \text{ értéke az } (x_1^0, x_2^0, \ldots, x_n^0) \text{ közelítésnél számolandó.}\right)$ Ha az egyenlet megoldása δx_2^1, akkor legyen

$$x_1^2 = x_1^1 + c_1^1 \delta x_2^1,$$

$$x_2^2 = x_2^0 + \delta x_2^1.$$

Ezután az

$$f_3(x_1^2+c_1^2\delta x_3, x_2^2+c_2^2\delta x_3, x_3^0+\delta x_3, x_4^0, \ldots, x_n^0) = 0$$

egyenletet oldjuk meg δx_3-ra, és közben feltesszük, hogy

$$df_1 = \frac{\partial f_1}{\partial x_1}\delta x_1+\frac{\partial f_1}{\partial x_2}\delta x_2+\frac{\partial f_1}{\partial x_3}\delta x_3 \equiv 0$$

$$df_2 = \frac{\partial f_2}{\partial x_1}\delta x_1+\frac{\partial f_2}{\partial x_2}\delta x_2+\frac{\partial f_2}{\partial x_3}\delta x_3 \equiv 0,$$

ahonnan a

$$\begin{bmatrix}\delta x_1\\ \delta x_2\end{bmatrix} = -\begin{bmatrix}\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2}\\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2}\end{bmatrix}^{-1}\begin{bmatrix}\frac{\partial f_1}{\partial x_3}\\ \frac{\partial f_2}{\partial x_3}\end{bmatrix}\cdot\delta x_3 = \begin{bmatrix}c_1^2\cdot\delta x_3\\ c_2^2\cdot\delta x_3\end{bmatrix}$$

összefüggés alapján számolhatjuk a c_1^2 és c_2^2 konstansokat.

Általában megoldjuk az

$$f_{k+1}(\mathbf{x}^k+\mathbf{q}^{k+1}\cdot\delta x_{k+1}) = 0$$

egyenletet δx_{k+1}-re $(k=0, 1, \ldots, n-1)$, ahol

$$\mathbf{x}^k = (x_1^k, \ldots, x_k^k, x_{k+1}^0, \ldots, x_n^0),$$

$$\mathbf{q}^{k+1} = \begin{bmatrix}\mathbf{p}^k\\ 1\end{bmatrix},\quad \mathbf{p}^k = -\mathbf{A}_k^{-1}\frac{\partial \mathbf{f}}{\partial x_{k+1}},\quad \mathbf{A}_k = \left\|\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right\|, \tag{1.2}$$

$$\mathbf{y} = (x_1, x_2, \ldots, x_k).$$

(A módszer használatához fel kell tennünk, hogy az $\mathbf{A}_k$ mátrixok nemszingulárisak.) $\mathbf{x}^0$-ból kiindulva $\mathbf{x}^n$ az első iteráció eredménye. Ezután $\mathbf{x}^n$-ből kiindulva az iteráció (többször is) megismételhető.

2. Lineáris eset

Ha az egyenletrendszer lineáris, azaz

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}-\mathbf{b} = \mathbf{0},$$

akkor az iterációs eljárás tetszőleges $\mathbf{x}^0$-ból kiindulva egy lépésben szolgáltatja a pontos megoldást, és a megoldás menete megegyezik a rendszámnöveléses mátrix-inverzió menetével (lásd [1], [2]). Ennek lépései a következők. Legyen

$$\mathbf{A}_{k+1} = \left\|\begin{matrix}\mathbf{A}_k & \boldsymbol{\omega}_k\\ \mathbf{v}_k^T & a_k\end{matrix}\right\|$$

és keressük az $\mathbf{A}_{k+1}$ inverzét a következő alakban:

$$\mathbf{A}_{k+1}^{-1} = \begin{Vmatrix} \mathbf{P}_k & \mathbf{r}_k \\ \mathbf{q}_k^T & b_k \end{Vmatrix},$$

ahol

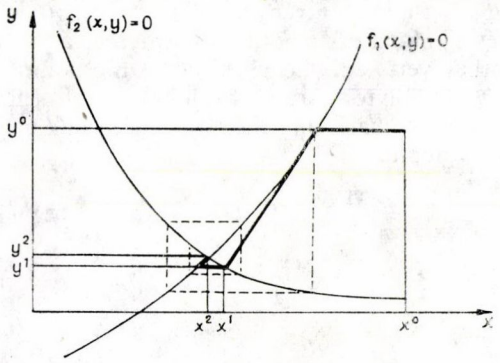
$$b_k = 1/c_k,\ c_k = a_k - \mathbf{v}_k^T \mathbf{A}_k^{-1} \boldsymbol{\omega}_k,$$

$$\mathbf{r}_k = -b_k \mathbf{A}_k^{-1} \boldsymbol{\omega}_k,\ \mathbf{q}_k^T = -b_k \mathbf{v}_k^T \mathbf{A}_k^{-1},$$

$$\mathbf{P}_k = \mathbf{A}_k^{-1} + b_k \mathbf{A}_k^{-1} \boldsymbol{\omega}_k \mathbf{v}_k^T \mathbf{A}_k^{-1}.$$

3. Geometriai szemléltetés

Minthogy az iterációs eljárás lineáris esetben a rendszámnöveléses mátrixinverziót adja, ezért az eljárás tekinthető annak nemlineáris esetre való általánosításának. Az eljárás első lépésben egy változóval (x_1-gyel) dolgozik, megold egy egyismeretlenes ($f_1=0$) egyenletet. A második lépésben is egy egyismeretlenes egyenletet old meg ($f_2=0$) a második (x_2) változóban, de úgy, hogy közben az első (x_1) megvál-

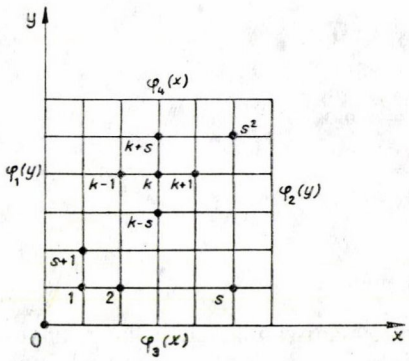


1. ábra

tozását is figyelembe veszi azáltal, hogy az első egyenlet bal oldalának megváltozása közelítőleg 0 maradjon ($df_1=0$). Általában minden lépésben egy nemlineáris egyenletet ($f_{k+1}=0$) old meg az x_{k+1} megváltozásában, de közben figyelembe veszi a többi x_i, $i \leqq k$ változó megváltozását is azáltal, hogy megköveteli, hogy a megelőző egyenletek differenciáljai ne változzanak, azaz $df_i=0$, $i \leqq k$. A módszer minden egyes lépésben a változók eggyel magasabb dimenziós terében dolgozik, vagyis a megoldás közben a változók terének a dimenziója lépésről lépésre eggyel növekszik. Ez indokolja a „rendszámnövelés" vagy „dimenzió kiterjesztés" elnevezést.

Kétváltozós esetben a módszert az 1. ábra szemlélteti.

Az 1. ábrán a szaggatott vonalak jelzik a *Gauss—Seidel módszerrel,* míg a vastagon kihúzott vonalak az általunk ajánlott módszerrel kapott közelítések alakulását. Az ábra szemlélteti, hogy a *Gauss—Seidel módszernél* több lépésre van szükség, azaz a nem lineáris egyenletet többször kell megoldani, mint az általunk ajánlott módszernél. Így az általunk ajánlott eljárás bonyolult rendszerek esetében előnyösebb, mint a *Gauss—Seidel eljárás.* Az iterációs eljárás konvergenciájával és a konvergencia rendjének vizsgálatával egy későbbi cikkben kívánunk foglalkozni.



2. ábra

4. Nemlineáris peremfeladat számolása

A módszer alkalmazásaként számoljuk ki numerikusan a

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = g(x, y, u) \tag{4.1}$$

nemlineáris egyenlet megoldását négyzeten, adott peremfeltételek mellett.

Tegyük fel, hogy a peremfeladatnak van megoldása, és a megoldás kiszámításával foglalkozunk.

Osszuk fel a négyzetet mindkét irányban $s+1$ részre (h a lépésköz, s^2 a belső rácspontok száma), és alkalmazzuk az ötpontos differencia sémát. Legyen az u függvény k-adik rácspontban vett i-edik közelítése u_k^i. A rácspontokat (a 2. ábrán látható módon) számozzuk be sorfolytonosan balról jobbra, alulról felfelé, akkor a k-adik rácsponthoz felírható az 1. szakaszban ismertetett eljárásnak megfelelően a következő nemlineáris differenciaegyenlet:

$$u_{k-s}^{k-1} + c_{k-s}\delta u_k + u_{k-1}^{k-1} + c_{k-1}\delta u_k + u_{k+1}^0 + u_{k+s}^0 - 4(u_k^0 + \delta u_k) =$$

$$= h^2 g(x_k, y_k, u_k^0 + \delta u_k),$$

vagy átrendezett formában

$$f_k = h^2 g(x_k, y_k, u_k^0 + \delta u_k) + (4 - c_{k-s} - c_{k-1})\delta u_k - \tag{4.2}$$

$$- (u_{k-s}^{k-1} + u_{k-1}^{k-1} + u_{k+1}^0 + u_{k+s}^0 - 4u_k^0) = 0,$$

ami nemlineáris egyenlet a δu_k ismeretlenre.

Ha (4.2) megoldása $\delta\bar{u}_k$, akkor legyen

$$u_k^k = u_k^0 + \delta\bar{u}_k,$$

$$u_i^k = u_i^{k-1} + c_i\delta\bar{u}_k, \quad i < k \text{ esetén,}$$

ahol a c_i együtthatók a

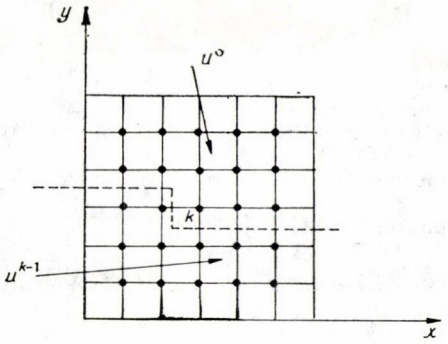
$$df_i = 0, \quad i < k$$

egyenletekből számolhatók.

Ha a perem melletti rácspontokra írjuk fel a differencia egyenletet, akkor az egyenletben szereplő peremben levő rácspontokban az adott függvényérték helyettesítendő.

A peremfeladat megoldásának k-adik lépését a 3. ábra segítségével szemléltetjük. Legyen a k-adik iterációs lépés előtt a rácspontbeli függvényértékek közelítése u_i^{k-1}, $i=1, \ldots, k-1$ és u_i^0, $i=k, \ldots, s^2$. A $\delta u_k = u_k^k - u_k^0$ közelítést az $f_k(\delta u_k)=0$-ból számítjuk, majd ennek hatását minden lépésben korrigáljuk az u_i, $i<k$ függvényértékekre.

Az iterációt végrehajtva minden rácsponthoz az $\mathbf{u}^0$-ból kiindulva a peremfeladat újabb közelítését kapjuk. Az eljárást ezután újra ismételhetjük, amíg meg nem kapjuk a kellő pontosságot.



3. ábra

A vizsgált peremfeladat esetén az egyenletrendszerünk *Jacobi mátrixa* a következő lesz

$$\mathbf{A} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{B}_2 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{B}_{s-1} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} & \mathbf{B}_s \end{bmatrix},$$

ahol

$$\mathbf{B}_i = \begin{bmatrix} d_{(i-1)s+1} & 1 & \cdots & 0 & 0 \\ 1 & d_{(i-1)s+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & d_{(i-1)s+s-1} & 1 \\ 0 & 0 & \cdots & 1 & d_{is} \end{bmatrix},$$

$d_k = -\left[4 + h^2 \dfrac{\partial g(x_k, y_k, u_k^0)}{\partial(\delta u_k)}\right]$ és $\mathbf{I}$ az egységmátrix.

Az iterációs eljárás egyszeri végrehajtása s^2 darab nemlineáris egyenlet megoldását és még ugyanannyi műveletet igényel, mint egy lineáris peremfeladat megoldása. A számolás algoritmusa pedig mindössze annyiban változik, hogy a $\mathbf{B}_i$ mátrix főátlójában $d_k = -\left[4 + h^2 \dfrac{\partial g(x_k, y_k, u_k^0)}{\partial(\delta u_k)}\right]$ számolandó ki.

A (4.2) egyenlet megoldásához szükséges c_i együtthatókat az (1.2) képlet segítségével kaphatjuk, ami az $\mathbf{A}_k$ azaz az $\mathbf{A}$ mátrix bal felső, $k \times k$ méretű részének invertálását igényli. (A $\mathbf{q}^k$ komponensei a c_i együtthatók.)

Az $\mathbf{A}$ mátrixunk csak kevés nem 0 elemet tartalmaz, ezért annak invertálását az általános mátrixinvertálásnál kevesebb számolási művelettel végezhetjük el (lásd [1]).

(1.2) szerint a c_i komponensekből álló $\mathbf{q}^{k+1}$ vektorhoz a

$$\mathbf{p}^k = -\mathbf{A}_k^{-1} \frac{\partial \mathbf{f}}{\partial x_{k+1}}$$

szorzásra van szükség, de minthogy a (4.2) egyenletünk $\frac{\partial \mathbf{f}}{\partial x_{k+1}}$ vektora mindössze két helyen tartalmaz nem 0 komponenst (ezek egyek lesznek az utolsó előtti és az azt megelőző $s+1$-edik helyen), az $\mathbf{A}_k^{-1} \frac{\partial \mathbf{f}}{\partial x_{k+1}}$ szorzás az $\mathbf{A}_k^{-1}$ megfelelő két oszlopának összegezését jelenti.

A fentiek alapján az algoritmus gépi programja nagyon könnyen elkészíthető. Ha (4.1)-ben $g \equiv 0$, akkor az $\mathbf{A}$ nem válik szinguláris.

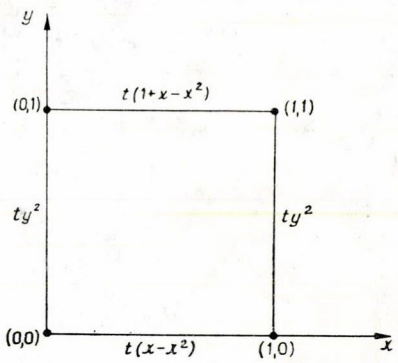
5. Próbaszámolások

A $\Delta u = u^2$ és a $\Delta u = u^3$ egyenletet oldjuk meg egységnégyzetre a 4. ábrán látható peremfeltételek mellett.

A t paraméter értékét $t = 1, 10, 100$ és 1000-nek választottuk ($\Delta u = u^3$ egyenletet $t = 10\,000$-re is megoldottuk). A t nagy értékére igen erős a nemlinearitás az egyenletekben. Az egyenleteket az összehasonlítás kedvéért megoldottuk az

$$\Delta u^{(k)} = u^{(k-1)} u^{(k)} \quad \text{és} \quad \Delta u^{(k)} = (u^{(k-1)})^2 u^{(k)} \tag{5.1}$$

iterációs eljárásokkal is. A számolásokat a CDC 3300-as gépen végeztük FORTRAN nyelven írt programokkal.



4. ábra

A számolások eredményeit az alábbi táblázatokban foglaljuk össze. (A két módszerrel kapott eredmények 7 tizedesjegyig megegyeztek.)

Kezdeti közelítésnek $\mathbf{u}^0 = 0$ (2. táblázatban), illetve a $\Delta u = 0$ egyenlet megoldását (1. táblázatban) vettük. A kétféle kiindulás a konvergenciát nem befolyásolta lényegesen. (A $\Delta u = 0$ megoldása ugyanazon programmal elvégezhető mint $\Delta u = u^2$ vagy $\Delta u = u^3$ megoldása.)

Az iterációk számának és a számolási időknek az összehasonlítását a 3. és 4. táblázatba foglaltuk össze. (A táblázatok számértékei a CDC CPU idejei secundum-ban.) A számolási pontosság 7 tizedesjegy volt.

A számolásokat a dolgozat által ajánlott módszerrel elvégeztük $s = 5$ és $s = 10$-re is. Az iterációk száma lényegesen nem változott.

1. TÁBLÁZAT. $\Delta u = u^2$ megoldása, ha $s = 3$

u_i \ t	1	10	100	1000
u_1	,243 363	2,046 243	11,826 848	50,979 725
u_2	,303 128	2,491 520	13,467 604	53,907 845
u_3	,243 363	2,046 243	11,826 848	50,979 725
u_4	,424 027	3,455 148	17,581 932	62,444 325
u_5	,481 516	3,761 574	16,552 741	45,300 417
u_6	,424 027	3,455 148	17,581 932	62,444 325
u_7	,732 468	6,258 901	36,268 412	147,203 01
u_8	,789 377	6,528 820	34,704 071	130,663 16
u_9	,732 468	6,258 901	36,268 412	147,203 01

2. TÁBLÁZAT. $\Delta u = u^3$ megoldása, ha s = 3

u_i \ t	1	10	100	1000	10000
u_1	,246 476	1,700 401	6,090 583	15,216 36	33,888 78
u_2	,307 439	2,011 994	6,460 480	15,382 14	33,942 92
u_3	,246 476	1,700 401	6,090 583	15,216 36	33,888 78
u_4	,492 402	2,596 892	7,022 563	15,680 67	34,084 69
u_5	,488 618	2,656 221	5,513 646	8,569 05	12,142 10
u_6	,429 402	2,596 892	7,022 563	15,680 67	34,084 69
u_7	,737 464	4,625 510	13,131 496	29,913 15	65,210 78
u_8	,795 520	4,590 419	12,025 007	26,858 60	58,338 43
u_9	,737 464	4,625 510	13,131 496	29,913 15	65,210 78

3. TÁBLÁZAT. $\Delta u = \dot{u}^2$, $s = 3$

	t	1	10	100	1000
Iterációs lépések száma	cikk által ajánlott módszer	3	4	5	6
	(5.1) képlettel	6	14	40	129
CPU idők	cikk által ajánlott módszer	,4	,5	,8	1,1
	(5.1) képlettel	1	2	5	17

4. TÁBLÁZAT. $\Delta u = u^3$, $s = 3$

	t	1	10	100	1000	10000
Iterációs lépések száma	cikk által ajánlott módszer	4	6	7	6	5
	(5.1) képlettel	7	96	*	*	*
CPU idők	cikk által ajánlott módszer	0,6	1,0	1,3	1,4	1,5
	(5.1) képlettel	1	13			

* Az (5.1) iterációval különböző kezdőértékekből kiindulva sem kaptunk eredményt.

6. A módszer használata, ha a rendszer sok lineáris egyenletet tartalmaz

Legyen az (1.1) rendszerünk $r<n$ egyenlete lineáris. Rendezzük (1.1)-et úgy, hogy először a lineáris, majd a nemlineáris egyenletek szerepeljenek, azaz

$$f_i(\mathbf{x}) = 0, \quad i \leqq r < n\text{-re lineáris}$$

$$f_i(\mathbf{x}) = 0, \quad r < i \leqq n\text{-re nemlineáris.}$$

Ebben az esetben a módszer különösen előnyösen alkalmazható. Invertáljuk a rendszer $\mathbf{A}$ *Jacobi mátrix*ának $(r\times r)$-es, bal-felső $\mathbf{A}_{rr}$ részét (ami nem függ a változóktól). Az $\mathbf{x}$ első r komponenséből álló vektort jelöljük $\mathbf{y}$-nal. Az $\mathbf{x}^0$-ból kiindulva kiszámítjuk a $\mathbf{z}=\mathbf{A}_{rr}^{-1}\mathbf{y}$ vektort, és az (1.2) képleteket csak $k=r, r+1, \ldots, n$-re kell végrehajtani, $k=r$ esetén $\mathbf{z}$-ből kiindulva. ($\mathbf{z}$ is r elemű vektor.) Az iterációs eljárás újabb ismétlésénél már nem kell $\mathbf{A}_{rr}^{-1}$-et újra kiszámolni, és ezáltal, ha r nagy, jelentős számolási munkát takarítunk meg.

IRODALOM

[1] GERGELY, J., „Numerikus módszerek sparse mátrixokra", *MTA SZTAKI Tanulmány*, **26** (1974).
[2] Фаддеев, Д. К. и Фаддеева, В. Н., *Вычислительные методы линейной алгебры* (Физматгиз, Москва, 1963).

(Beérkezett: 1975. június 9.)

DR. GERGELY JÓZSEF
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST I., ÚRI U. 49.

A NUMERICAL SOLUTION OF THE SYSTEMS OF NONLINEAR EQUATIONS

J. GERGELY

The paper presents the following algorithm to solve the system of equations $\mathbf{f}(\mathbf{x})=0$, $\mathbf{x}\in R^n$, $\mathbf{f}\in R^n$. Be $\mathbf{x}^0=(x_1^0, \ldots, x_n^0)$ and $\mathbf{x}^k=(x_1^k, \ldots x_k^k, x_{k+1}^0, \ldots, x^0)$ for $k=1, \ldots, n$.

Let us consider $\delta x_{k+1}^k = x_{k+1}^{k+1} - x_{k+1}^0$ $(k=0, \ldots, n-1)$ as a solution of equation $f_{k+1}(\mathbf{x}^k + \mathbf{q}^{k+1}\delta x_{k+1}^k)=0$, where

$$\mathbf{q}^{k+1} = \begin{bmatrix} \mathbf{p}^k \\ 1 \end{bmatrix}, \quad \mathbf{p}^k = -\mathbf{A}_k^{-1} \frac{\delta f}{\delta x_{k+1}}\bigg|_{\mathbf{x}=\mathbf{x}^0}, \quad A_k = \left\| \frac{\delta \mathbf{f}}{\delta \mathbf{y}}\bigg|_{\mathbf{x}=\mathbf{x}^0} \right\|$$

for $k=1, \ldots, n-1$, $q^1=1$ and $\mathbf{y}$ is the vector containing the first k cordinate of $\mathbf{x}$.

The method is very well applicable for solving numerically nonlinear elliptic partial differential equations.

The paper considers the equation $\Delta u=g(x, y, u)$ in a square domain and demonstrates some numerical results.

Alkalmazott Matematikai Lapok **2** *(1976) 135—143*

A KVÁZI-NEWTON MÓDSZEREK EGY ÚJ HÁROMPARAMÉTERES OSZTÁLYA*

ABAFFY JÓZSEF
Budapest

A cikkben a *kvázi-Newton módszerek* olyan új háromparaméteres osztályát adjuk meg, amely tartalmazza a *Broyden osztályt* és ekvivalens *Huang* ötparaméteres osztályával. Igazoljuk, hogy a javasolt módszerosztály kvadratikusan konvergens és *Broyden* értelemben stabilis.

1. Bevezetés

A *kvázi-Newton módszerek* a feltétel nélküli függvényminimalizálás témakörében igen fontos szerepet játszanak. Ezért az utóbbi években számos *kvázi-Newton módszert* dolgoztak ki, és megindultak a kutatások az ismert eljárások egységes tárgyalására. Az egységes tárgyalás olyan kvadratikusan konvergens, egy vagy több paraméteres iterációs séma megadását jelenti, amelyből a paraméterek alkalmas megválasztásával az ismert *kvázi-Newton módszerek* adódnak ([2], [3]). Elsőként C. G. BROYDEN 1967-ben a szimmetrikus *kvázi-Newton módszerek* egy egyparaméteres osztályát definiálta ([1]). 1970-ben pedig H. J. HUANG olyan ötparaméteres általános sémát adott meg, amelyből az ismert szimmetrikus *kvázi-Newton módszereken* kívül a nem-szimmetrikusok is következnek ([2]). A jelen dolgozatban megadunk egy olyan háromparaméteres általános iterációs sémát, amely a következő tulajdonságokkal rendelkezik:

a) Kiadja az ismert szimmetrikus és nem-szimmetrikus módszereket.

b) Kvadratikus befejezésű.

c) A *kvázi-Newton módszerek* ugyanazon osztályát fedi le, mint Huang ötparaméteres osztálya.

d) Tartalmazza a *Broyden-osztályt.*

e) Stabil.

2. A háromparaméteres iterációs séma levezetése

Tekintsük az

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c \tag{2.1}$$

kvadratikus függvényt, ahol $\mathbf{A}$ $(n\times n)$-es pozitív definit szimmetrikus mátrix és $\mathbf{x}$, $\mathbf{b}\in R^n$, $c\in R^1$. Azt mondjuk, hogy a $\mathbf{p}_i\in R^n$ $(\mathbf{p}_i\neq\mathbf{0}, i=1, 2, \ldots, n)$ vektorok $\mathbf{A}$

* A cikk 2. és 3. fejezete elhangzott 1974. július 31-én a dublini „Numerikus Analízis" konferencián.

— konjugált irányok, ha

$$\mathbf{p}_i^T \mathbf{A} \mathbf{p}_j = \begin{cases} =0, & i \neq j, \\ \neq 0, & i = j, \end{cases} \quad i, j = 1, 2, \ldots, n. \tag{2.2}$$

Az $\mathbf{A}$ mátrix pozitív definitsége miatt a $\mathbf{p}_i$ konjugált irányok lineárisan függetlenek. Az általános sémát a következő alakban keressük:

$$\mathbf{p}_i = -H_i^T \mathbf{g}_i, \quad i = 1, 2, \ldots, \tag{2.3}$$

$$\mathbf{y}_i = \mathbf{g}_{i+1} - \mathbf{g}_i, \quad i = 1, 2, \ldots, \tag{2.4}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i \quad i = 1, 2, \ldots, \tag{2.5}$$

ahol az α_i paraméterre teljesül

$$f(\mathbf{x}_{i+1}) = \min_{\alpha \in R^1} f(\mathbf{x}_i + \alpha \mathbf{p}_i), \quad i = 1, 2, \ldots, \tag{2.6}$$

$$\mathbf{H}_{i+1} = \mathbf{A}_i(\mathbf{H}_i, \mathbf{y}_i, \mathbf{p}_i, \alpha_i, \beta_i, \delta_i, \varrho_i), \quad i = 1, 2, \ldots, \tag{2.7}$$

ahol $\mathbf{H}_i$ $(n \times n)$-es mátrixsorozat, $\mathbf{g}_i = \operatorname{grad} f(\mathbf{x}_i)$ és $\alpha_i, \beta_i, \delta_i, \varrho_i$ olyan konstansok, amelyek közül α_i-t a (2.6) vonal menti minimalizálás meghatározza $(i = 1, 2, \ldots)$. Olyan $\{\mathbf{A}_i\}$ mátrixsorozatot választunk, amelyre a (2.3)—(2.7) algoritmus véges, legfeljebb n lépésben megadja $f(\mathbf{x})$ minimumát, és a (2.3) által definiált $\mathbf{p}_i$ irányokra a (2.2) feltétel teljesül.

Korlátozzuk az $\mathbf{A}_i$ meghatározását a következő módon:

$$\mathbf{A}_i(\mathbf{H}_i, \mathbf{y}_i, \mathbf{p}_i, \alpha_i, \beta_i, \delta_i, \varrho_i)\mathbf{y}_i = \alpha_i \varrho_i \mathbf{p}_i, \quad i = 1, 2, \ldots, \tag{2.8}$$

és keressük a következő alakban

$$\mathbf{A}_i = \mathbf{H}_i + \mathbf{B}_i(\mathbf{H}_i, \mathbf{y}_i, \mathbf{p}_i, \alpha_i, \beta_i, \delta_i, \varrho_i), \quad i = 1, 2, \ldots. \tag{2.9}$$

Megjegyezzük, hogy a (2.8) feltétel még szimmetrikus esetben sem határozza meg egyértelműen az $\mathbf{A}_i$ mátrixot.

A (2.8) és (2.9) kifejezésekből azt kapjuk, hogy

$$\mathbf{B}_i(\mathbf{H}_i, \mathbf{y}_i, \mathbf{p}_i, \alpha_i, \beta_i, \delta_i, \varrho_i)\mathbf{y}_i = \alpha_i \varrho_i \mathbf{p}_i - \mathbf{H}_i \mathbf{y}_i, \quad i = 1, 2, \ldots. \tag{2.10}$$

Olyan $\mathbf{z}_i, \mathbf{q}_i \in R^n$ vektorokat választva, amelyekre

$$\mathbf{z}_i^T \mathbf{y}_i = 1, \ \mathbf{q}_i^T \mathbf{y}_i = 1, \quad i = 1, 2, \ldots, \tag{2.11}$$

azt kapjuk, hogy

$$\mathbf{B}_i(\mathbf{H}_i, \mathbf{y}_i, \mathbf{p}_i, \alpha_i, \beta_i, \delta_i, \varrho_i) = \alpha_i \varrho_i \mathbf{p}_i \mathbf{q}_i^T - \mathbf{H}_i \mathbf{y}_i z_i^T, \tag{2.12}$$

ami (2.10) általánosításának tekinthető.

Legyenek a $\mathbf{q}_i, \mathbf{z}_i \in R^n$ irányok a következő alakúak

$$\mathbf{q}_i^T = \beta_i \mathbf{p}_i^T + \frac{1 - \beta_i \mathbf{p}_i^T \mathbf{y}_i}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i} \mathbf{y}_i^T \mathbf{H}_i, \quad i = 1, 2, \ldots, \tag{2.13}$$

$$\mathbf{z}_i^T = -\alpha_i \delta_i \mathbf{p}_i^T + \frac{\alpha_i \delta_i \mathbf{p}_i^T \mathbf{y}_i + 1}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i} \mathbf{y}_i^T \mathbf{H}_i, \quad i = 1, 2, \ldots. \tag{2.14}$$

Ekkor azt kapjuk, hogy

$$\mathbf{H}_{i+1} = \mathbf{H}_i + \alpha_i \varrho_i \beta_i \mathbf{p}_i \mathbf{p}_i^T + \alpha_i \varrho_i \frac{1 - \beta_i \mathbf{p}_i^T \mathbf{y}_i}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i} \mathbf{p}_i \mathbf{y}_i^T \mathbf{H}_i + \tag{2.15}$$

$$+ \alpha_i \delta_i \mathbf{H}_i \mathbf{y}_i \mathbf{p}_i^T - \frac{\alpha_i \delta_i \mathbf{p}_i^T \mathbf{y}_i + 1}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i} \mathbf{H}_i \mathbf{y}_i \mathbf{y}_i^T \mathbf{H}_i, \quad i = 1, 2, \ldots.$$

A (2.3)—(2.6) és (2.15) kifejezések együttesen definiálják az általános háromparaméteres iterációs sémát.

3. Kvadratikus befejezés

Először a kvadratikus befejezés fogalmát definiáljuk, amely eltér az analízisben szokásos kvadratikus konvergencia fogalmától.

3.1. DEFINÍCIÓ. Azt modjuk, hogy a (2.3)—(2.6) és (2.15) által meghatározott algoritmus kvadratikus befejezésű, ha a (2.1) által definiált $f(\mathbf{x})$ függvény minimumát legfeljebb n iterációs lépésben meghatározza.

Egy iterációs lépésen a (2.3)—(2.6) és a (2.15) kifejezések egyszeri kiszámítását értjük.

Érvényes a következő tétel.

3.1. TÉTEL. A (2.3)—(2.6) és (2.15) relációk által meghatározott algoritmus kvadratikus befejezésű tetszőleges paraméterválasztás mellett.

Bizonyítás. Először belátjuk, hogy

$$\mathbf{p}_i^T \mathbf{A} \mathbf{p}_j = 0, \quad 1 \leqq i, j < k, \quad i \neq j, \tag{3.1}$$

$$\mathbf{H}_k \mathbf{A} \alpha_i \mathbf{p}_i = \alpha_i \varrho_i \mathbf{p}_i \quad 1 \leqq i < k. \tag{3.2}$$

Az igazolást teljes indukcióval végezzük. A $k=2$ esetben

$$\mathbf{H}_2 \mathbf{A} \alpha_1 \mathbf{p}_1 = \mathbf{H}_1 \mathbf{y}_1 + \alpha_1 \varrho_1 \beta_1 \mathbf{p}_1 \mathbf{p}_1^T \mathbf{y}_1 + \alpha_1 \varrho_1 \frac{1 - \beta_1 \mathbf{p}_1^T \mathbf{y}_1}{\mathbf{y}_1^T \mathbf{H}_1 \mathbf{y}_1} \mathbf{p}_1 \mathbf{y}_1^T \mathbf{H}_1 \mathbf{y}_1 + \tag{3.3}$$

$$+ \alpha_1 \delta_1 \mathbf{H}_1 \mathbf{y}_1 \mathbf{p}_1^T \mathbf{y}_1 - \frac{\alpha_1 \delta_1 \mathbf{p}_1^T \mathbf{y}_1 + 1}{\mathbf{y}_1^T \mathbf{H}_1 \mathbf{y}_1} \mathbf{H}_1 \mathbf{y}_1 \mathbf{y}_1^T \mathbf{H}_1 \mathbf{y}_1 = \alpha_1 \varrho_1 \mathbf{p}_1,$$

ahol kihasználtuk, hogy

$$\mathbf{A} \alpha_i \mathbf{p}_i = \mathbf{y}_i, \quad i = 1, 2, \ldots, n. \tag{3.4}$$

A (3.1) indukciós feltevést $j=1, i=2$-re igazoljuk. A (2.3) relációt felhasználva

$$\mathbf{p}_2^T \mathbf{A} \mathbf{p}_1 = -\mathbf{g}_2^T \mathbf{H}_2 \mathbf{A} \mathbf{p}_1 = -\mathbf{g}_2^T \mathbf{p}_1 \varrho_1 = 0,$$

amennyiben α_1-et (2.6) szerint határozzuk meg. Tegyük fel, hogy (3.1) és (3.2) igazak k-ig. A (3.1) képlet esetén azt kell belátnunk, hogy

$$\mathbf{p}_k^T \mathbf{A} \mathbf{p}_j = 0, \quad 1 \leqq j \leqq k-1.$$

Minthogy a (3.2) relációt feltéve

$$\mathbf{p}_k^T \mathbf{A}\mathbf{p}_j = -\mathbf{g}_k^T \mathbf{H}_k \mathbf{A}\mathbf{p}_j = -\frac{\varrho_i}{\alpha_i} \mathbf{g}_k^T \mathbf{p}_j,$$

azt kell csupán belátnunk, hogy $\mathbf{g}_k^T \mathbf{p}_j = 0$, $1 \leqq j \leqq k-2$, mert $\mathbf{g}_k^T \mathbf{p}_{k-1}$ helyessége (2.6)-ból következik.

A (2.5) kifejezés miatt írhatjuk, hogy

$$\mathbf{x}_k = \mathbf{x}_{j+1} + \sum_{l=j+1}^{k-1} \alpha_l \mathbf{p}_l,$$

amiből

$$\mathbf{A}\mathbf{x}_k - \mathbf{b} = \mathbf{A}\mathbf{x}_{j+1} - \mathbf{b} + \sum_{l=j+1}^{k-1} \mathbf{A}\alpha_l \mathbf{p}_l.$$

Innen adódik, hogy

$$\mathbf{g}_k^T \mathbf{p}_j = \mathbf{g}_{j+1}^T \mathbf{p}_j + \sum_{l=j+1}^{k-1} \alpha_l \mathbf{p}_l^T \mathbf{A}\mathbf{p}_j = 0,$$

ahol kihasználtuk a (3.1) indukciós feltevést és azt, hogy (2.6) miatt $\mathbf{g}_{j+1}^T \mathbf{p}_j = 0$.

A (3.2) kifejezés érvényességét $k = k+1$-re két lépésben látjuk be. Legyen először $k = k+1$ és $i = k$. Ekkor (3.3)-hoz hasonlóan

$$\mathbf{H}_{k+1} \mathbf{A}\alpha_k \mathbf{p}_k = \alpha_k \varrho_k \mathbf{p}_k. \tag{3.5}$$

Legyen $1 \leqq i < k$. Ekkor a (2.15) összefüggést felhasználva

$$\mathbf{H}_{k+1} \mathbf{A}\alpha_i \mathbf{p}_i = \mathbf{H}_{k+1} \mathbf{y}_i = \mathbf{H}_k \mathbf{y}_i + \alpha_k \varrho_k \beta_k \mathbf{p}_k \mathbf{p}_k^T \mathbf{y}_i + \alpha_k \varrho_k \frac{1 - \beta_k \mathbf{p}_k^T \mathbf{y}_k}{\mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_k} \mathbf{p}_k \mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_i + \tag{3.6}$$

$$+ \alpha_k \delta_k \mathbf{H}_k \mathbf{y}_k \mathbf{p}_k^T \mathbf{y}_i - \frac{\alpha_k \delta_k \mathbf{p}_k^T \mathbf{y}_k + 1}{\mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_k} \mathbf{H}_k \mathbf{y}_k \mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_i.$$

Az $\mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_i$ és a $\mathbf{p}_k^T \mathbf{y}_i$ kifejezésre a következőket kapjuk. Egyrészt

$$\mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_i = \alpha_i \varrho_i \mathbf{y}_k^T \mathbf{p}_i = \alpha_i \alpha_k \varrho_i \mathbf{p}_k^T \mathbf{A}\mathbf{p}_i = 0,$$

másrészt

$$\mathbf{p}_k^T \mathbf{y}_i = \alpha_i \mathbf{p}_k^T \mathbf{A}\mathbf{p}_i = 0.$$

A (3.6) kifejezés tehát a következő alakra hozható

$$\mathbf{H}_{k+1} \mathbf{A}\alpha_i \mathbf{p}_i = \mathbf{H}_k \mathbf{y}_i.$$

A (3.2) indukciós feltevés, (3.4) és (3.5) alapján

$$\mathbf{H}_{k+1} \mathbf{A}\alpha_i \mathbf{p}_i = \alpha_i \varrho_i \mathbf{p}_i, \quad 1 \leqq i \leqq k,$$

amivel az indukciós lépést igazoltuk.

Ebből a 3.1. definíció értelmében vett kvadratikus befejezés már adódik, mert a $\mathbf{p}_i \neq 0 \in R^n$, $i = 1, 2, \ldots, n$ konjungált irányok kifeszítik a teret, és az irányukban végzett minimalizálással az n-edik lépésben $f(\mathbf{x})$ minimuma adódik.

4. Az általános iterációs séma és más kvázi-Newton módszer osztályok

Ebben a pontban az általános iterációs séma és a *Huang*, valamint a *Broyden osztály* kapcsolatával foglalkozunk. *Huang* általános osztályát a (2.3)—(2.6) és a

(4.1)
$$\mathbf{H}_{i+1} = \mathbf{H}_i + \varrho \frac{\alpha_i \mathbf{p}_i (C_1 \alpha_i \mathbf{p}_i + C_2 \mathbf{H}_i^T \mathbf{y}_i)^T}{(C_1 \alpha_i \mathbf{p}_i + C_2 \mathbf{H}_i^T \mathbf{y}_i)^T \mathbf{y}_i} - \frac{\mathbf{H}_i \mathbf{y}_i (K_1 \alpha_i \mathbf{p}_i + K_2 \mathbf{H}_i^T \mathbf{y}_i)^T}{(K_1 \alpha_i \mathbf{p}_i + K_2 \mathbf{H}_i^T \mathbf{y}_i)^T \mathbf{y}_i}, \quad i = 1, 2, \ldots$$

mátrixsorozat definiálja.

A (2.15) és (4.1) kifejezések összehasonlításából a

(4.2)
$$\frac{\varrho C_1 \alpha_i}{(C_1 \alpha_i \mathbf{p}_i + C_2 \mathbf{H}_i^T \mathbf{y}_i)^T \mathbf{y}_i} = \beta_i \varrho_i, \quad i = 1, 2, \ldots,$$

(4.3)
$$\frac{\varrho C_2}{(C_1 \alpha_i \mathbf{p}_i + C_2 \mathbf{H}_i^T \mathbf{y}_i)^T \mathbf{y}_i} = \varrho_i \frac{1 - \beta_i \mathbf{p}_i^T \mathbf{y}_i}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i}, \quad i = 1, 2, \ldots,$$

(4.4)
$$\frac{K_1}{(K_1 \alpha_i \mathbf{p}_i + K_2 \mathbf{H}_i^T \mathbf{y}_i)^T \mathbf{y}_i} = -\delta_i, \quad i = 1, 2, \ldots,$$

(4.5)
$$\frac{K_2}{(K_1 \alpha_i \mathbf{p}_i + K_2 \mathbf{H}_i^T \mathbf{y}_i)^T \mathbf{y}_i} = \frac{\delta_i \alpha_i \mathbf{p}_i^T \mathbf{y}_i + 1}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i}, \quad i = 1, 2, \ldots$$

egyenlőségek adódnak.

A továbbiakban igazoljuk, hogy a (4.1) és a (2.15) által meghatározott módszer osztályok ekvivalensek.

Ha $\varrho^2 + C_1^2 + C_2^2 > 0$ és $C_1 \alpha_i \mathbf{p}_i^T \mathbf{y}_i \neq -C_2 \mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i$ (különben (4.2), (4.3) nevezője zérussá válna), akkor ϱ_i, β_i-re következő egyenletrendszer adódik:

(4.6)
$$\beta_i \varrho_i = \frac{\varrho C_1 \alpha_i}{(C_1 \alpha_i \mathbf{p}_i + C_2 \mathbf{H}_i^T \mathbf{y}_i)^T \mathbf{y}_i}, \quad i = 1, 2, \ldots,$$

(4.7)
$$\varrho_i \frac{1 - \beta_i \mathbf{p}_i^T \mathbf{y}_i}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i} = \frac{\varrho C_2}{(C_1 \alpha_i \mathbf{p}_i + C_2 \mathbf{H}_i^T \mathbf{y}_i)^T \mathbf{y}_i}, \quad i = 1, 2, \ldots.$$

A fenti (4.6), (4.7) egyenletrendszer a β_i, ϱ_i ismeretlenekre egyértelműen megoldható, nevezetesen

(4.8)
$$\varrho_i = \frac{\varrho C_1 \alpha_i \mathbf{p}_i^T \mathbf{y}_i + \varrho C_2 \mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i}{(C_1 \alpha_i \mathbf{p}_i + C_2 \mathbf{H}_i^T \mathbf{y}_i)^T \mathbf{y}_i} = \varrho, \quad i = 1, 2, \ldots$$

és

(4.9)
$$\beta_i = \frac{C_1 \alpha_i}{(C_1 \alpha_i \mathbf{p}_i + C_2 \mathbf{H}_i^T \mathbf{y}_i)^T \mathbf{y}_i}, \quad i = 1, 2, \ldots.$$

A (4.5) reláció (4.4)-ből következik, ezért a (4.4) által definiált megoldás (4.5)-öt kielégíti.

Fordítva, tegyük fel, hogy ϱ_i, β_i adott konstansok, és meghatározandók (4.2) és (4.3)-ból a ϱ, C_1 és C_2 ismeretlenek, úgy, hogy $\varrho^2+C_1^2+C_2^2>0$ és $C_1\alpha_i\mathbf{p}_i^T\mathbf{y}_i\neq$ $\neq -C_2\mathbf{y}_i^T\mathbf{H}_i\mathbf{y}_i$. Minthogy két egyenletünk van három ismeretlen meghatározására, a $\varrho\equiv\varrho_i$ választással élhetünk. Ha $\varrho_i\equiv 0$, akkor C_1 és C_2 tetszőleges, mert a (4.1) kifejezés második tagja eltűnik, a (2.15) kifejezés 2. és 3. tagjához hasonlóan. Így a (4.2) és (4.3) kifejezések helyébe a

$$\frac{C_1\alpha_i}{(C_1\alpha_i\mathbf{p}_i+C_2\mathbf{H}_i^T\mathbf{y}_i)\mathbf{y}_i}=\beta_i,\quad i=1,2,\ldots, \tag{4.10}$$

$$\frac{C_2}{(C_1\alpha_i\mathbf{p}_i+C_2\mathbf{H}_i^T\mathbf{y}_i)\mathbf{y}_i}=\frac{1-\beta_i\mathbf{p}_i^T\mathbf{y}_i}{\mathbf{y}_i^T\mathbf{H}_i\mathbf{y}_i},\quad i=1,2,\ldots \tag{4.11}$$

kifejezések lépnek. A $\beta_i\equiv 0$ esetben $C_1=0$, $C_2=1$ a megoldás. Minthogy a $C_1\equiv$ $\equiv C_2\equiv 0$ és a $C_1\alpha_i\mathbf{p}_i^T\mathbf{y}_i=-C_2\mathbf{y}_i^T\mathbf{H}_i\mathbf{y}_i$ eseteket kizártuk, a következő két egyenlet adódik:

$$C_1(\alpha_i-\alpha_i\mathbf{p}_i^T\mathbf{y}_i\beta_i)=C_2\mathbf{y}_i^T\mathbf{H}_i\mathbf{y}_i\beta_i,\quad i=1,2,\ldots, \tag{4.12}$$

$$C_2\beta_i\mathbf{p}_i^T\mathbf{y}_i=C_1\frac{\alpha_i\mathbf{p}_i^T\mathbf{y}_i(1-\beta_i\mathbf{p}_i^T\mathbf{y}_i)}{\mathbf{y}_i^T\mathbf{H}_i\mathbf{y}_i},\quad i=1,2,\ldots. \tag{4.13}$$

Minthogy a két egyenlet egymásnak konstans-szorosa, (4.12) alapján tetszőleges β_i konstanshoz C_1 és C_2 meghatározható úgy, hogy $C_1\equiv C_2\equiv 0$ és a $C_1\alpha_i\mathbf{p}_i^T\mathbf{y}_i=$ $=-C_2\mathbf{y}_i^T\mathbf{H}_i\mathbf{y}_i$ relációk nem állnak fenn.

Legyen végül δ_i adott érték, és határozzuk meg K_1-et és K_2-t úgy, hogy $K_1^2+K_2^2>0$ és

$$K_1\alpha_i\mathbf{p}_i^T\mathbf{y}_i\neq -K_2\mathbf{y}_i^T\mathbf{H}_i\mathbf{y}_i.$$

A $\vartheta_i=-\alpha_i\delta_i$ jelöléssel (4.4) és (4.5) átmegy a következő két egyenletbe:

$$\frac{K_1\alpha_i}{(K_1\alpha_i\mathbf{p}_i+K_2\mathbf{H}_i^T\mathbf{y}_i)^T\mathbf{y}_i}=\vartheta_i,\quad i=1,2,\ldots,$$

$$\frac{K_2}{(K_1\alpha_i\mathbf{p}_i+K_2\mathbf{H}_i^T\mathbf{y}_i)^T\mathbf{y}_i}=\frac{1-\vartheta_i\mathbf{p}_i^T\mathbf{y}_i}{\mathbf{y}_i^T\mathbf{H}_i\mathbf{y}_i},\quad i=1,2,\ldots,$$

amelyek a (4.10) és a (4.11) egyenletekkel azonosak, tehát K_1 és K_2 alkalmas módon meghatározható.

A (2.15) kifejezésből következik, hogy szimmetrikus *kvázi-Newton módszert* kapunk, ha

$$\alpha_i\delta_i=\alpha_i\varrho_i\frac{1-\beta_i\mathbf{p}_i^T\mathbf{y}_i}{\mathbf{y}_i^T\mathbf{H}_i\mathbf{y}_i} \tag{4.14}$$

minden i-re teljesül. Innen

$$\alpha_i\varrho_i\beta_i=\frac{\alpha_i\varrho_i-\alpha_i\delta_i\mathbf{y}_i^T\mathbf{H}_i\mathbf{y}_i}{\mathbf{p}_i^T\mathbf{y}_i},\quad i=1,2,\ldots,$$

amelynek megfelelően (2.15) a következőképpen alakul:

$$\mathbf{H}_{i+1} = \mathbf{H}_i + \frac{\alpha_i \varrho_i - \alpha_i \delta_i \mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i}{\mathbf{p}_i^T \mathbf{y}_i} \mathbf{p}_i \mathbf{p}_i^T + \alpha_i \delta_i (\mathbf{p}_i \mathbf{y}_i^T \mathbf{H}_i + \mathbf{H}_i \mathbf{y}_i \mathbf{p}_i^T) - \tag{4.15}$$

$$- \frac{\alpha_i \delta_i \mathbf{p}_i^T \mathbf{y}_i + 1}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i} \mathbf{H}_i \mathbf{y}_i \mathbf{y}_i^T \mathbf{H}_i, \quad i = 1, 2, \ldots.$$

Ha $\varrho = 1$ és $\delta_i = -\vartheta_i$ $(i = 1, 2, \ldots)$, akkor a *kvázi—Newton módszerek Broyden osztályát* kapjuk.

5. Az ismert kvázi-Newton módszerek mint az általános séma speciális esetei

Ebben a fejezetben megmutatjuk, hogyan adódnak az ismert *kvázi-Newton módszerek* a (2.3)—(2.6) és a (2.15) által definiált sémából.

A $\varrho_i = 1$ eset. Már láttuk, hogy $\varrho_i = 1$ esetén a $\delta_i = -\vartheta_i$ választással a *Broyden osztály* adódik. Minthogy (4.14) miatt a *Broyden osztály* maximális, ezért a szimmetrikus módszerek tárgyalásától eltekinthetünk.

A $\delta_i = -\dfrac{1}{\alpha_i \mathbf{p}_i^T \mathbf{y}_i}$ és $\beta_i = \dfrac{\alpha_i}{\alpha_i \mathbf{p}_i^T \mathbf{y}_i}$ $(i = 1, 2, \ldots)$ választással a *McCormick algoritmus* adódik:

$$\mathbf{H}_{i+1} = \mathbf{H}_i + \frac{(\alpha_i \mathbf{p}_i - \mathbf{H}_i \mathbf{y}_i) \alpha_i \mathbf{p}_i^T}{\alpha_i \mathbf{p}_i^T \mathbf{y}_i}, \quad i = 1, 2, \ldots.$$

Legyen $\delta_i = 0$ és $\beta_i = 0$, akkor a *Pearson algoritmust* kapjuk:

$$\mathbf{H}_{i+1} = \mathbf{H}_i + \frac{(\alpha_i \mathbf{p}_i - \mathbf{H}_i \mathbf{y}_i) \mathbf{y}_i^T \mathbf{H}_i}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i}, \quad i = 1, 2, \ldots.$$

A $\varrho_i = 0$ eset. A (2.15) formula alapján a következő adódik:

$$\mathbf{H}_{i+1} = \mathbf{H}_i + \alpha_i \delta_i \mathbf{H}_i \mathbf{y}_i \mathbf{p}_i^T - \frac{\alpha_i \delta_i \mathbf{p}_i^T \mathbf{y}_i + 1}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i} \mathbf{H}_i \mathbf{y}_i \mathbf{y}_i^T \mathbf{H}_i, \quad i = 1, 2, \ldots. \tag{5.1}$$

Kiemelve ebből az osztályból a $\delta_i = \pm 1$, $\delta_i = 0$ és $\delta_i = -\dfrac{1}{\alpha_i \mathbf{p}_i^T \mathbf{y}_i}$ eseteket, két új *kvázi-Newton módszert* kapunk.

Ha $\delta_i = -1$, akkor az alábbi módszert kapjuk:

$$\mathbf{H}_{i+1} = \mathbf{H}_i - \mathbf{H}_i \mathbf{y}_i \alpha_i \mathbf{p}_i^T + \frac{\alpha_i \mathbf{p}_i^T \mathbf{y}_i - 1}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i} \mathbf{H}_i \mathbf{y}_i \mathbf{y}_i^T \mathbf{H}_i, \quad i = 1, 2, \ldots. \tag{5.2}$$

Ha $\delta_i = 0$, akkor a HUANG [1] cikkben közölt V. algoritmust kapjuk:

$$\mathbf{H}_{i+1} = \mathbf{H}_i - \frac{\mathbf{H}_i \mathbf{y}_i \mathbf{y}_i^T \mathbf{H}_i}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i}, \quad i = 1, 2, \ldots.$$

Legyen most $\delta_i = 1$. Ekkor a következő módszer adódik:

(5.3)
$$\mathbf{H}_{i+1} = \mathbf{H}_i + \mathbf{H}_i \mathbf{y}_i \alpha_i \mathbf{p}_i^T - \frac{\alpha_i \mathbf{p}_i^T \mathbf{y}_i + 1}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i} \mathbf{H}_i \mathbf{y}_i \mathbf{y}_i^T \mathbf{H}_i, \quad i = 1, 2, \ldots.$$

Ha most $\delta_i = -\frac{1}{\alpha_i \mathbf{p}_i^T \mathbf{y}_i}$ $(i = 1, 2, \ldots)$, akkor a Huang VI. algoritmusát kapjuk:

(5.4)
$$\mathbf{H}_{i+1} = \mathbf{H}_i - \frac{\mathbf{H}_i \mathbf{y}_i \alpha_i \mathbf{p}_i^T}{\alpha_i \mathbf{p}_i^T \mathbf{y}_i}, \quad i = 1, 2, \ldots.$$

Végül pedig ha $\delta_i = -\frac{1}{\alpha_i \mathbf{p}_i^T \mathbf{y}_i - \mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i}$, akkor HUANG VII. algoritmusát kapjuk:

$$\mathbf{H}_{i+1} = \mathbf{H}_i - \frac{\mathbf{H}_i \mathbf{y}_i (\alpha_i \mathbf{p}_i - \mathbf{H}_i^T \mathbf{y}_i)^T}{(\alpha_i \mathbf{p}_i - \mathbf{H}_i^T \mathbf{y}_i)^T \mathbf{y}_i}, \quad i = 1, 2, \ldots.$$

6. Stabilitási problémák

A 2. pontban leírtakból adódik, hogy a stabilitási definíciónak a következő két feltételt kell tartalmaznia:

6.1. DEFINÍCIÓ. ([1]) Egy szimmetrikus *kvázi-Newton módszerről* azt mondjuk, hogy stabil, ha teljesíti a következő két feltételt

a) $\mathbf{H}_i$ pozitív definit mátrix minden i-re,

b) $\mathbf{z}_i^T \mathbf{y}_i = 1$ és $\mathbf{q}_i^T \mathbf{y}_i = 1$, minden i-re.

Az a) feltételre vonatkozóan a következő tételt bizonyítjuk.

6.1. TÉTEL. Ha $\mathbf{H}_1$ szimmetrikus pozitív definit mátrix és $\varrho_i > 0$, $\delta_i \leqq 0$ minden i-re, akkor $\mathbf{H}_i$ szimmetrikus pozitív definit mátrix $(i = 1, 2, \ldots)$.

Bizonyítás. Minthogy $\mathbf{H}_1$ szimmetrikus pozitív definit mátrix, elegendő bizonyítani, hogy $\mathbf{H}_i$ pozitív definitségéből $\mathbf{H}_{i+1}$ pozitív definit volta következik.

A (4.15) kifejezés alapján

$$\mathbf{x}^T \mathbf{H}_{i+1} \mathbf{x} = \mathbf{x}^T \mathbf{H}_i \mathbf{x} + \frac{\alpha_i \varrho_i - \alpha_i \delta_i \mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i}{\mathbf{p}_i^T \mathbf{y}_i} \mathbf{x}^T \mathbf{p}_i \mathbf{p}_i^T \mathbf{x} + \alpha_i \delta_i (\mathbf{x}^T \mathbf{p}_i \mathbf{y}_i^T \mathbf{H}_i \mathbf{x} + \mathbf{x}^T \mathbf{H}_i \mathbf{y}_i \mathbf{p}_i^T \mathbf{x}) -$$

$$- \frac{\alpha_i \delta_i \mathbf{p}_i^T \mathbf{y}_i + 1}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i} \mathbf{x}^T \mathbf{H}_i \mathbf{y}_i \mathbf{y}_i^T \mathbf{H}_i \mathbf{x}, \quad i = 1, 2, \ldots.$$

Minthogy $\mathbf{H}_i$ pozitív definit mátrix (és egyben szimmetrikus is), létezik a következő felbontása

$$\mathbf{H}_i = \mathbf{D} \cdot \mathbf{D}^T.$$

Bevezetve az $\mathbf{l}=\mathbf{D}^T\mathbf{x}$, $\mathbf{v}=\mathbf{D}^T\mathbf{y}_i$, $\mathbf{w}=\mathbf{D}^T\mathbf{g}_i$ jelöléseket, felhasználva a $\mathbf{p}^T\mathbf{x}=-\mathbf{g}_i^T\mathbf{H}_i\mathbf{x}$ egynelőséget, kapjuk, hogy

$$\mathbf{x}^T\mathbf{H}_{i+1}\mathbf{x} = (\mathbf{l},\mathbf{l})-\frac{\alpha_i\varrho_i-\alpha_i\delta_i(\mathbf{v},\mathbf{v})}{(\mathbf{w},\mathbf{v})}(\mathbf{l},\mathbf{w})^2-2\alpha_i\delta_i(\mathbf{l},\mathbf{w})\cdot(\mathbf{l},\mathbf{v})- \tag{6.1}$$

$$-\frac{1-\alpha_i\delta_i(\mathbf{w},\mathbf{v})}{(\mathbf{v},\mathbf{v})}(\mathbf{v},\mathbf{l})^2=\frac{(\mathbf{l},\mathbf{l})(\mathbf{v},\mathbf{v})-(\mathbf{v},\mathbf{l})^2}{(\mathbf{v},\mathbf{v})}-\frac{\alpha_i\varrho_i-\alpha_i\delta_i(\mathbf{v},\mathbf{v})}{(\mathbf{w},\mathbf{v})}(\mathbf{l},\mathbf{w})^2-$$

$$-2\alpha_i\delta_i(\mathbf{l},\mathbf{w})(\mathbf{l},\mathbf{v})+\frac{\alpha_i\delta_i(\mathbf{w},\mathbf{v})}{(\mathbf{v},\mathbf{v})}(\mathbf{v},\mathbf{l})^2,\quad i=1,2,\ldots,$$

ahol $(\mathbf{x},\mathbf{y})$ a skalárszorzatot jelöli.

A (6.1) egyenlőség jobb oldalának utolsó három tagja pozitív, ha

$$\frac{-\alpha_i\varrho_i+\alpha_i\delta_i(\mathbf{v},\mathbf{v})}{(\mathbf{w},\mathbf{v})}+\frac{\alpha_i\delta_i(\mathbf{w},\mathbf{v})}{(\mathbf{v},\mathbf{v})}>-2\alpha_i\delta_i,\quad i=1,2,\ldots. \tag{6.2}$$

Belátjuk, hogy ha a tétel feltételei teljesülnek, akkor a (6.2) egyenlőtlenség igaz. Először megmutatjuk, hogy $(\mathbf{w},\mathbf{v})<0$, ugyanis

$$(\mathbf{w},\mathbf{v})=\mathbf{g}_i^T\mathbf{H}_i\mathbf{y}_i=\mathbf{g}_i^T\mathbf{H}_i(\mathbf{g}_{i+1}-\mathbf{g}_i)=-\mathbf{g}_i^T\mathbf{H}_i\mathbf{g}_i<0,\quad i=1,2,\ldots.$$

Az $a=\frac{(\mathbf{w},\mathbf{v})}{(\mathbf{v},\mathbf{v})}$ $(a<0)$ jelöléssel (6.2) a következőképpen alakul

$$-\frac{\alpha_i\varrho_i}{(\mathbf{w},\mathbf{v})}>-\alpha_i\delta_i\left(a+\frac{1}{a}+2\right),\quad i=1,2,\ldots.$$

A baloldalra a $\varrho_i>0$ feltétel miatt $-\frac{\alpha_i\varrho_i}{(\mathbf{w},\mathbf{v})}>0$ teljesül. A jobboldalra viszont $-\alpha_i\delta_i\geqq 0$ és $a+\frac{1}{a}+2\leqq 0$ miatt $-\alpha_i\delta_i\left(a+\frac{1}{a}+2\right)\leqq 0$, amivel előbbi állításunkat beláttuk, A (6.1) kifejezés jobb oldalának első tagja a *Schwarz egyenlőtlenség* miatt nem negatív, amivel a tételt beláttuk.

A 6.1. definíció b) része a (2.13) és a (2.14) összefüggések értelmében nyilvánvalóan teljesül.

IRODALOM

[1] BROYDEN, C. G., "Quasi-Newton Methods and their Application to Function Minisation", *Math. of Comp.* **21** (1967) 368—381.

[2] HUANG, H. Y., "Unified Approach to Quadratically Convergent Algorithms for Function Minisation", *Journal of Optimization Theory and Application* **5** (1970) 405—423.

[3] PEARSON, J. D., "Variable metric methods of minimisation", *Comp. J.* **12** (1969) 171—178.

(Beérkezett: 1975. augusztus 25.)

ABAFFY JÓZSEF
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST I., ÚRI U. 49.

A THREE PARAMETRIC CLASS OF *QUASI-NEWTON METHODS*

J. ABAFFY

In this paper we introduce a new class of *Quasi-Newton methods* depending on three parameters only. This class includes the *Broyden class* and it is equivalent to the *Huang class* which depends on five parameters. This class has quadratic termination property and it is stable in the *Broyden sense.*

Alkalmazott Matematikai Lapok **2** *(1976) 145—149*

Hírek és közlemények

III. MATEMATIKAI PROGRAMOZÁSI TÉLI ISKOLA
Mátrafüred, 1975. január 31—február 6.

A *Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézete* az előző évekhez hasonlóan 1975 telén is *Matematikai Programozási Téli Iskolát* rendezett *Mátrafüreden*, a *Magyar Tudományos Akadémia* üdülőjében.

A Téli Iskola vezetője és tudományos programjának kialakítója DR. PRÉKOPA ANDRÁS egyetemi tanár volt, a szervezési teendőket pedig e sorok írója látta el az Intézetben működő szervező iroda közreműködésével.

A program összeállításakor törekedtünk arra, hogy a Téli Iskola jellege változatlan maradjon, hogy az előadások az előadók saját új eredményein kívül a témakör áttekintését is tartalmazzák, s az előadásokon kívül közös programok segítsék elő a szakmai és baráti kapcsolatok elmélyítését.

A résztvevők számát a nagy érdeklődés ellenére sem növeltük, mert véleményünk szerint ez a Téli Iskola munkájának hatékonyságát csökkentette volna.

A Téli Iskolán való részvételre és előadás tartására szóló meghívásunkat 20 külföldi kolléga fogadta el, rajtuk kívül 9 külföldi hallgatója volt az Iskolának. A magyar résztvevők száma 51 volt, ebből 19-en intézetünk dolgozói.

A tudományos program 23 előadást tartalmazott, (20 külföldi, 1 magyar meghívott és 2 intézeti előadást terveztünk), ezek közül 4 elmaradt. Egy-egy előadás időtartama 1 óra volt. Az előadások között 15 perc szünetet tartottunk a felmerülő kérdések kötetlen megbeszélésére. A legtöbb előadás angolul hangzott el, a továbbiak németül, egy pedig oroszul. Az alábbiakban rövid áttekintést adunk az elhangzott előadásokról.

BELL, D. E. *(IIASA) "The Group Approach to Integer Programming: An outline of Recent Progress"* c. előadásában olyan egészértékű lineáris programozás feladattal foglalkozott, melyben az együtthatók egészek. Összefoglalta a diszkrét programozási irodalomban csoportelméleti módszer néven ismert eredményeket. Mint ismeretes, a feladat feltételei gyengíthetők úgy, hogy egy csoport elemeiből mint csúcsokból és élekből képzett gráfon kell egy legrövidebb utat keresni. Ez a gyengítés beágyazható egy korlátozás és szétválasztás típusú algoritmusba, amely megadja az egzakt optimumot. A témával magyar szerzők is több ízben foglalkoztak már.

DEUMLICH, R. és ELSTER, K. H. *(NDK) "On the Theory of Conjugate Functions"* című előadásban a szerzők a konjugált függvények egy általánosítását vizsgálták. A konjugált függvények szoros kapcsolatban vannak a $\varphi_0 : z=\frac{1}{2}\mathbf{x}^T\mathbf{x}$ hiperparaboloid polaritási tulajdonságaival. A fenti általánosításban φ_0-t tetszőleges nem degenerált másodrendű felülettel helyettesítik. Definiálták konvex halmaznak valamely P pontra vonatkozó epigráfját, majd halmazoknak a P pontra vonatkozó konvexitását. Valamely függvényt konvexnek neveznek egy P pontra vonatkozóan, ha a gráfja konvex halmaz erre a P pontra nézve. Az általuk definiált konvexitási fogalom vizsgálata után bebizonyították annak a jól ismert tételnek az általánosítását, mely szerint zárt konvex függvény gráfja az általa majorizált affin függvények felső burkolója. Ezután egy a P pontra vonatkozóan konvex függvény konjugáltját definiálták, és bebizonyították néhány tételt az új konjugáltsági relációra.

DUPACOVA, J. *(Csehszlovákia) "On minimax Decision Rule in Stochastic Linear Programming"*. Olyan lineáris programozási feladatokban, amelyekben a paraméterek valószínűségi változók, de ezek együttes valószínűségeloszlása ismeretlen, célszerű az ún. *minimax elv* alkalmazása. J. DUPACOVA előadásában a vonatkozó matematikai jellegű eredmények egy jó áttekintését adta. Az eredményeket az előadó, JOSIFESCU és THEODORESCU érték el.

JEREMIN, I. I. és MAZUROV, V. D. *(Szovjetunió) "Synthesis of Non-stationary Processes of Mathematical Programming, Containing Pattern Recognation"* előadása az $x_{t+1} \in F_t x_t$, vagy $(x_t, x_{t+1}) \in T_t$ alakú folyamatokkal foglalkozott, ezek alkalmazását tekintette olyan matematikai programozási problémák megoldására, melyek információkomponense időben változik, és rosszul definiált elemeket tartalmaz. Foglalkozott a fenti folyamatok konkrét realizációival és tervezési, irányítási problémákban való alkalmazásaival.

GUDDAT, J. *(NDK), "On the Qualitative Stability in the Nonlinear Parametric Programming and Some Applications"* előadásában nemlineáris parametrikus programozási problémák stabilitás vizsgálatával foglalkozott. Attekintette a konvex kvadratikus programozási probléma lokális stabilitásával kapcsolatos eredményeit, majd alkalmazta ezeket vektorminimum problémákra és nemlineáris programozási problémák diszkretizálására *(Karbs és Burns értelemben)*.

HOLLATZ, H. *(NDK) »Ein analitisch-geometrischer Zugang zu notwendigen Bedingungen bei Minimax-Aufgaben«* előadás célja DEMJANOV és MALOZEMOV *»Einführung in Minimax«* című könyvében szereplő tételek egységes tárgyalása volt. Egy igen általános, konvex kúpokra, illetve ezek polárisaira vonatkozó lemmából indult ki, amelyet a konvex analízis eszközeivel bizonyított. Ezután ennek speciális eseteként megkapta a *Fritz John tételt* és a *Kuhn—Tucker tételt*. A kiinduló lemma felhasználásával bebizonyította a fenti könyvben szereplő, az optimalitás szükséges feltételeit adó tételeket, illetve ezeket magában foglaló, általános tételt.

HUARD, P. *(Franciaország) »Synthese et agrégation des methodes en Programmation mathématique«* előadása P. HUARD és a *Lille-i egyetemen* dolgozó munkatársai által az utóbbi néhány évben elért optimalizálási eredményeket foglalta össze. Az elméleti eredmények főként az általános algoritmusokkal az egydimenziós optimalizálással és algoritmusok kompozíciójával kapcsolatosak. A számítógépes realizációk a nemlineáris és a diszkrét programozási algoritmusokkal kapcsolatosak.

JEWELL, W. S. *(IIASA) "Introduction to isotonic Estimation"*. A szerző egy az optimális becslések elméletébe tartozó problémát vizsgál. Modelljében a becslések „jóságára" vonatkozó szokásos kritériumok — az eltérések négyzetösszege legyen minimális, a súlyozott abszolút eltérések összege, az abszolút eltérések maximuma legyen minimális — kielégítésén túl egy részben rendezési relációt is ki kell elégítenie a becsléseknek. E részben-rendezés által megengedett „legjobb" becslés meghatározásának problémáját fogalmazza át a dolgozat optimális költségű kapacitásos folyamfeladat megoldására. A szerző a folyamfeladatot megoldó algoritmus lépéseit fogalmazza vissza, és értelmezi a kitűzött feladatra.

KÉRI, G. *(MTA SZTAKI) "A tableau filling problem and its possible application to matrix inversion."* LP kódok hatékonyságát nagymértékben meghatározza az alkalmazott invertálási technika. KÉRI GERZSON előadásában a lokális, illetve globális feltöltődés problémájával foglalkozott, ennek egy általánosítását tekintette véletlen, 0—1 értékeket tartalmazó $n \times m$-es mátrixok vizsgálatával. Az ezekkel kapcsolatos probléma: mennyi az olyan különböző $m \times n$-es 0—1-es mátrixok száma, melyben a sor-, illetve oszlop-összegek adottak. Vizsgálatai alapján a *Markovitz-féle számok* módosítását javasolta.

LOVÁSZ, L. *(József Attila Tudományegyetem) "Efficient algorithms: an approach by formal logic"* előadásában kombinatorikus problémák karakterizálhatóságát, „jó" algoritmussal való megoldhatóságát vizsgálta. Az általa felsorakoztatott példák sejtetik, hogy az adott problémákat karakterizáló nem-triviális szükséges és elégséges feltételek, illetve a problémák polinom algoritmussal való megoldhatósága szorosan összefügg. Az előadó a példaként mondott kombinatorikus feladatokat a formális logika nyelvén fogalmazza meg. A bevezetett logikai struktúrák vizsgálatával egységesen tárgyalhatók az olyan látszólag egymástól távoleső feladatok, mint adott gráfban *Hamilton kör* keresése, egy gráf síkba rajzolhatóságának eldöntése, vagy egy szám prímvoltának megállapítása. Az előadásban a tételeken túl az előadó 7 sejtése is ismertetésre került.

PRÉKOPA, A. *(MTA, SZTAKI) "Dynamic type stochastic programming models"* előadása a legfontosabb dinamikus típusú sztochasztikus programozási modellek összefoglaló ismertetését nyújtotta. Némely vonatkozásban új modellkonstrukciót is tartalmazott, továbbá újszerűek voltak a sztochasztikus és dinamikus modellek általános sémájával és konstrukciós elveivel kapcsolatos fejtegetések. Az ismertetett modellek a következők: kétlépcsős sztochasztikus programozási modell (DANTZIG—MADANSKY); ennek előzményei (DANTZIG—BEALE); további variánsai (PRÉKOPA); több lépcsős változata (DANTZIG); több lépcsős modell valószínűségi feltétel és büntetés szimultán alkal-

mazásával (PRÉKOPA—SZÁNTAI); valószínűség szekvenciális maximalizálása (PRÉKOPA); az *E*- és a *P*-modell (CHARNES—KIRBY); egy több periódusos és több cikkes modell (BEALE—FORREST—TAYLOR); kétszektoros több periódusos közgazdasági modell (TINTNER—RAGHAVAN); általános sztochasztikus és dinamikus modellséma (PRÉKOPA).

WEINERT, H. *(NDK), "On the Solution of Certain Multi-Parametric Linear Programming Problems"* előadásában korábbi eredményeire támaszkodva egy módszert adott, amellyel a különböző egy- és több paraméteres, konstans együtthatómátrixszal rendelkező lineáris programozási problémákra vonatkozó eredmények áttekinthetők. Néhány speciális esetre az ún. lokális stabilitási terület meghatározására szolgáló eljárások ötleteit is ismertette, s végül rámutatott arra, hogy az előadásban nem tárgyalt, nemlineáris paraméter függést tartalmazó problémákra hogyan alkalmazhatók az ismertetettek.

WINKLER, C. *(IIASA) "Basis Factorization for Block Angular Linear Programs"* előadásában először megmutatta, hogyan specializálható a szimplex módszer blokk anguláris szerkezetű lineáris programozási feladatok megoldására — a bázis blokk anguláris szerkezetének kihasználásával. Külön-külön foglalkozott az összekapcsoló feltételeket tartalmazó, az összekapcsoló változatokat tartalmazó és a kétszeresen összekapcsolt feladatok megoldásaival. Áttekintette az ismert dekompozíciós eljárásokat (eltekintve a *Dantzig—Wolfe módszertől*), s megmutatta, hogyan származtathatók ezek a fenti módszerből a bázisba bevonandó és a bázisból kilépő vektor kiválasztására alkalmazott különböző stratégiákkal.

ZIMMERMANN, K. *(Csehszlovákia) "Solution of Some Optimization Problems on the Extremal Algebra"* előadásában az ún. extremális algebrai struktúrák nyelvén megfogalmazható optimalizálási feladatokat vizsgálta. Az alaphalmaz algebrai tulajdonságait kihasználva algoritmust adott az ilyen típusú feladatok megoldására. Az ismertetett módszer különleges értéke, hogy — az ismert módszerekkel nehezen kezelhető — nem-konvex matematikai programozási problémák egy része is megfogalmazható az extremális algebrai struktúrák nyelvén. Az előadásban közölt algoritmus már konkrét feladat esetén — annak speciális tulajdonságait kihasználva — tovább csiszolható.

ZIONTS, S. és WALLENIUS, J. *(Svájc) "On Finding the Subset of Efficient Vectors of an Arbitrary Set of Vectors"* — egy olyan algoritmust adott vektorhalmaz efficiens vektorainak a kiválasztására, mely csekély váltottazással az „efficiens vektor" többféle definíciója esetén alkalmazható. Az algoritmus lineáris programozási feladatok megoldásán keresztül vezet az efficiens vektorok megtalálására.

A Téli Iskolán előadást tartott az *IIASA* 3 tudományos munkatársa is, akik az érdeklődésre való tekintettel egy kerekasztal beszélgetésben ismertették az *IIASA* munkáját, célkitűzéseit, felépítését stb.

A tudományos programon kívül egy félnapos buszkirándulást is szerveztünk, utolsó este pedig búcsúvacsora volt az üdülőben.

A Téli Iskola Szervezői úgy vélik, hogy a Téli Iskola munkája eredményes volt. Az eltelt három év alatt a *Mátrafüredi Téli Iskola* a hazai és nemzetközi Matematikai Programozás elismert és figyelemmel kísért eseménye lett.

Strazicky Beáta

A KORSZERŰ PROGRAMOZÁSI NYELVEK KRITÉRIUMAI

A *Magyar Tudományos Akadémia Számítástudományi Bizottsága* 1976. február 17-én kibővített ülést hívott össze néhány, a korszerű programozási nyelvekkel kapcsolatos aktuális kérdés megvitatására. A vitaülésen a *Bizottság* tagjain kívül mintegy 60 meghívott szakember is részt vett.

Napjainkban is születnek hazánkban különböző célokra programozási nyelvek. Szükség van-e ilyen fejlesztésekre? Milyen követelményeknek kell eleget tennie egy korszerű programozási nyelvnek? Merre halad a programozási nyelvek fejlődése? Ilyen és ezekhez hasonló kérdésekre kereste a *Bizottság* a választ.

A vitát ARATÓ MÁTYÁS, a matematikai tudományok doktora, a bizottság elnöke vezette.

A vitaülésre hat vitaindító referátum téziseit nyújtották be írásban. Ezeket a téziseket a meghívottak előre megkapták.

DÖMÖLKI BÁLINT: *„A programozási nyelvek ártalmasságáról (Programming Languages Considered Harmful...")* címmel tartotta meg referátumát. Az előadás címével a szerző azon írásművek hosszú sorára utalt, amelyek DIJKSTRA-nak a go to utasítás elleni első támadását követően a programozási nyelvek különböző elemei, illetve különböző programozási módszerek ellen irányultak. Az előadó úgy definiálta a programozási nyelveinket, mint azon korlátozások, megszorítások összességét, amelyeket a gépek (számítástechnikai rendszerek) mai intelligencia szintje állít a feladatok természetes módon történő meghatározásának útjába.

Szerinte a „software-krízis" alapproblémája az a nagyságrendi különbség, amely a feladatok bonyolultsági szintje és a megoldásnál használható gépek által közvetlenül megérthető eszközök szintje között fennáll. A „software-krízis" megoldásának kulcsát nem a programozási nyelvek, hanem a programozási módszertan fejlesztésében látja.

NÉMETI TIBOR: hasonló elveket fejtegetett. Véleménye szerint a problémát nem a programozási nyelvek sokfélesége jelenti, hanem a programozás módjának szabálytalansága, az „egyéni", mindenféle szabványosítástól való idegenkedő programozás. Ezért a hangsúlyt ma nem a korszerű programozási nyelvekre, hanem a korszerű programozásra kell helyezni. Ilyen korszerű programozási módszer a strukturált programozás.

FARKAS ERNŐ megállapította, hogy a jelenleg használatos programozási nyelvek általában nem a felhasználók igényeit tükrözik, hanem az ezeket létrehozó rendszerprogramozók matematikai, programozás elméleti felkészültségét és az ebből leszűrt esztétikai elveket. Ennek következménye az a helyzet, hogy a programok döntő része ma is FORTRAN-ban és COBOL-ban készül, holott nem ezeket tekintjük a legkorszerűbbeknek.

FARKAS ERNŐ szerint olyan új nyelvekre van szükség, amelyeket tudatosan terveznek, de a felhasználók számára.

„Minek abból még öt, amiből már van egy tucat?" — ez volt LŐCS GYULA előadásának mottója. Ugyanazt a programrészletet más-más nyelven megírva bizonyította, hogy a nyelvek architektúrája kísértetiesen hasonlít egymásra, a szerzők azonban gondosan ügyelnek arra, hogy az új nyelv kellően inkompatibilis legyen minden meglevővel. Az ötletek forrása — kevés számú kivételtől eltekintve — kimutathatóan az ALGOL68, az ALGOL60 és kisebb mértékben a PL/I.

Előadásának másik részében a strukturált programozási elvekkel ellentétben álló defaultokról mutatta meg, hogy ezek mennyire akadályozzák a strukturált programozás eredeti alapgondolatának, a jól áttekinthető programozási stílusnak az érvényre jutását.

GEHÉR ISTVÁN véleménye szerint a magasabb rendű nyelveknek rendelkezni kell olyan input utasítással, amely nemcsak adatstruktúrát, hanem egy bináris relokálható programot is be tud olvasni valamely előre deklarált vagy generált tömbbe, és rendelkeznie kell azzal a képességgel is, hogy a vezérlés erre a programrészletre adható legyen.

BEDŐ ÁRPÁD, LABORCZI ZOLTÁN, LANGER TAMÁS vitatéziseit BEDŐ ÁRPÁD mondta el.

Nézetük szerint a programozási nyelveknek nem a számológép programozását, hanem a feladatok jó megoldását kell egyszerűvé tenniük. Kérdéses az univerzális nyelvek létjogosultsága is. Nem nyelvekben, hanem rendszerekben kell gondolkodni.

Felvetették a két nyelv problémáját is: egy a program bejáratásához, egy az éles futtatáshoz.

A vitaindító referátumok után megrendezett vitában 11 felszólalás hangzott el.

Az első hozzászóló KALMÁR LÁSZLÓ professzor volt. Kifejtette, hogy a jelenlegi software problémák nemcsak a programozási módszer kifejlesztésével, hanem a gépek értelmi szintjének növelésével (formula vezérlésű számítógép) is megoldhatók.

A továbbiakban felszólalt ZSOMBOK ZOLTÁN, MÜNNICH ANTAL, SZEREDI PÉTER, DETTRICH ÁRPÁD, ASZALÓS JÁNOS, BÓKA PÉTER, HAVAS MIKLÓS, BACH IVÁN.

A felszólalások alapján a korszerű programozási nyelv legfontosabb kritériumait a következőképpen foglalhatjuk össze:

A korszerű rendszerprogramozási nyelv
— legyen egyszerű,
— világos szemantikájú és könnyen megérthető,
— késztesse a felhasználót áttekinthető programok írására,
— tegye lehetővé a program hibáinak gyors felderítését,
— minden általánosan elterjedt és viszonylag kis kapacitású univerzális számítógépre lehessen alkalmazni.

A *Számítástudományi Bizottság* a vita alapján a következőket állapította meg:

A fenti objektív kritériumok alapján a felszólalók általában más-más programozási nyelvet részesítették előnyben. Sok programozási nyelv azonos hatással rendelkezik, nagy különbség nincsen köztük, így nagy a megszokás hatása. Így a programozási nyelvek fejlődésében ugrásszerű változás nem várható.

A programozási nyelv, mint minden élő nyelv, fejlődik. A fejlődést az alkalmazási területek igényei kell, hogy diktálják. Célszerű azonban mérsékelni az új nyelvek beáramlását.

A „software-krízis" megoldásának kulcsa a programozási módszertan fejlesztésében van. Ennek eredményei alapján az a távolság, amelyet a feladatok bonyolultsági szintje és a használatos gépek értelmi szintje között meglevő nagy szintkülönbség jelent, több különböző programozási szint bevezetésével hidalható át. Minden egyes ilyen szint tulajdonképpen egy-egy absztrakt programozási nyelvet határoz meg. Ilyen szinteken keresztül kell eljutni a specifikációs szinttől az implementációs szintig.

Gálfi Zoltán

A kiadásért felel az Akadémiai Kiadó igazgatója
Műszaki Szerkesztő: Agócs András
A kézirat nyomdába érkezett: 1976. VIII. 6. Terjedelem: 13,3 (A/5) ív
76-3382 — Szegedi Nyomda — Felelős vezető: Dobó József igazgató

ÚTMUTATÁS A SZERZŐKNEK

Az Alkalmazott Matematikai Lapok csak magyar nyelvű dolgozatokat közöl. A kéziratok gépelését olyan formában kérjük, hogy minden gépelt oldal 25, egyenként átlag 50 betűhelyes sort tartalmazzon. A közlésre szánt dolgozatokat három példányban a felelős szerkesztő címére kell beküldeni:

Prékopa András, felelős szerkesztő, MTA SZTAKI
1502 Budapest XI., Kende u. 13—17.

A kéziratok szerkezeti felépítésének a következő követelményeket kell kielégíteni. A fejlécnek tartalmaznia kell a dolgozat címét, a szerző teljes nevét, valamint annak a városnak a nevét, ahol a szerző dolgozik. A fejléc után egy, képletet nem tartalmazó, legfeljebb 200 szóból álló kivonatot kell minden esetben megadni. A dolgozatot címmel ellátott szakaszokra kell bontani, és az egyes szakaszokat arab sorszámmal kell ellátni. Az esetleges bevezetésnek mindig az első szakaszt kell alkotnia. Az irodalomjegyzék mindig az utolsó szakasz kell hogy legyen, és azt nem kell sorszámmal ellátni. Az irodalomjegyzék után, a kézirat befejezéseképpen fel kell tüntetni a szerző teljes nevét és a munkahelye (illetve lakása) pontos postai címét. A dolgozatban előforduló képleteket szakaszonként újrakezdődően, a képlet előtt két zárójel közé írt kettős számozással kell azonosítani. Természetesen nem szükséges minden képletet számozással ellátni. Az esetleges definíciókat és tételeket (segédtételeket és lemmákat) ugyancsak szakaszonként újrakezdődő, kettős számozással kell ellátni. Kérjük a szerzőket, hogy ezeket, valamint a tételek bizonyítását a szövegben kellő módon emeljék ki. Minden dolgozathoz csatolni kell egy angol, német, francia vagy orosz nyelvű, külön oldalra gépelt összefoglalót. Amennyiben lehetséges, kérjük a nyomtatás számára különösen nehézkes matematikai jelölések használatának az elkerülését.

A dolgozat ábráit és az esetleges lábjegyzeteket a dolgozat végén, különálló lapokon kérjük beküldeni. Mind az ábrákat, mind a lábjegyzeteket a dolgozat szakaszokra bontásától független, folytatólagos arab sorszámozással kell ellátni. Az ábrák elhelyezését a dolgozat megfelelő helyén, széljegyzetként feltüntetett, ábraazonosító sorszámokkal kell megadni. A lábjegyzetekre a dolgozaton belül az azonosító sorszám felső indexkénti használatával lehet hivatkozni.

Az irodalmi hivatkozások formája a következő. Minden hivatkozást fel kell sorolni a dolgozat végén található irodalomjegyzékben, a szerzők, illetve társszerzők esetén az első szerző neve szerinti alfabetikus sorrendben úgy, hogy külön, de folytatólagos sorszámozású listát alkossanak a latin és a cirill betűs nevű szerzők műveire vonatkozó hivatkozások, és mindkét részben a megfelelő alfabetikus sorrend legyen kialakítva. A folyóiratban megjelent cikkekre [1], a könyvekre [5], a kötetben megjelent dolgozatokra [4], a disszertációkra [3] és a gépi program leírásokra [2] a következő minta szerint kell hivatkozni:

[1] Farkas, J., »Über die Theorie der einfachen Ungelichungen«, *Journal für die reine und angewandte Mathematik* **124** (1902) 1—27.

[2] Kéri, G., „DUALSIMP", rutin a CDC 3300-as gépekre (Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézete, CDC 3300 felhasználói ismertetők 2. 1973. május) 19—20.

[3] Prékopa, A., „Sztohasztikus rendszerek optimalizálási problémáiról", doktori értekezés. Magyar Tudományos Akadémia, Budapest, 1970.

[4] Prabhu, N. U., "Recent research on the ruin problem of collective risk theory", in: *Inventory Control and Water Storage* Ed. A. Prékopa (János Bolyai Mathematical Society and North-Holland Publishing Company, Amsterdam—London, 1973) 221—228.

[5] Zoutendijk, G., *Methods of Feasible Directions* (Elsevier Publishing Company, Amsterdam and New York, 1960).

A dolgozatok szövegében az irodalmi hivatkozás számait szögletes zárójelben kell megadni, mint például [5] vagy [4, 76—78]. A szerzők a dolgozatukról 100 darab különlenyomatot kapnak, ezek költsége — nyomott oldalanként 25 forint — a szerzői díjat terheli.

Ára: 30,— Ft

INDEX: 26498
ISSN 0133—3399

TARTALOMJEGYZÉK

INDEX

Alkalmazott matematikai lapok

1976/3-4

A MAGYAR TUDOMÁNYOS AKADÉMIA
MATEMATIKAI ÉS FIZIKAI TUDOMÁNYOK
OSZTÁLYÁNAK KÖZLEMÉNYEI

2.
KÖTET

AKADÉMIAI KIADÓ, BUDAPEST

A MAGYAR TUDOMÁNYOS AKADÉMIA

MATEMATIKAI ÉS FIZIKAI TUDOMÁNYOK OSZTÁLYÁNAK

ALKALMAZOTT MATEMATIKAI LAPJA

II. kötet 3—4. szám

Szerkesztőség: 1502 Budapest XI., Kende u. 13—17.

Kiadóhivatal: 1055 Budapest V., Alkotmány u. 21.

Az Alkalmazott Matematikai Lapok változó terjedelmű füzetekben jelenik meg, és olyan eredeti tudományos cikkeket publikál, amelyek a gyakorlatban, vagy más tudományokban közvetlenül felhasználható új matematikai eredményt tartalmaznak, illetve már ismert, de színvonalas matematikai apparátus újszerű és jelentős alkalmazását mutatják be. A folyóirat közöl cikk formájában megírt, új tudományos eredménynek számító programokat, és olyan, külföldi folyóiratban már publikált dolgozatokat, amelyek magyar nyelven történő megjelentetése elősegítheti az elért eredmények minél előbbi, széles körű hazai felhasználását.

A folyóirat feladata a Magyar Tudományos Akadémia III. (Matematikai és Fizikai) Osztályának munkájára vonatkozó közlemények, könyvismertetések stb. publikálása is.

Kéziratok a következő címre küldendők:

Prékopa András, felelős szerkesztő
1502 Budapest XI., Kende u. 13—17.

Ugyanerre a címre küldendő minden szerkesztőségi levelezés.

Közlésre el nem fogadott kéziratokat a szerkesztőség lehetőleg visszajuttat a szerzőhöz, de a beküldött kéziratok megőrzéséért vagy továbbításáért felelősséget nem vállal.

Az Alkalmazott Matematikai Lapok előfizetési ára kötetenként 60 forint. Belföldi megrendelések az Akadémiai Kiadó, 1055 Budapest V., Alkotmány u. 21. címen (pénzforgalmi jelzőszám 215—11 488), külföldi megrendelések a Kultúra Külkereskedelmi Vállalat, H-1389 Budapest, Pf. 149. címen (pénzforgalmi jelzőszám 218—10 990) lehetségesek.

A Magyar Tudományos Akadémia III. (Matematikai és Fizikai) Osztálya a következő idegen nyelvű folyóiratokat adja ki:

1. Acta Mathematica Hungaricae,
2. Acta Physica Hungaricae,
3. Studia Scientiarum Mathematicarum Hungarica.

Alkalmazott Matematikai Lapok **2** *(1976) 151—155*

KALMÁR LÁSZLÓ
(1905–1976)

Súlyos veszteség érte a magyar tudományt, a magyar matematikát és közelebbről az Alkalmazott Matematikai Lapokat, 1976. augusztus 2-án váratlanul elhunyt Kalmár László nyugalmazott egyetemi tanár a Magyar Tudományos Akadémia rendes tagja, az Alkalmazott Matematikai Lapok főszerkesztője.

Kalmár László a Somogy megyei Edde községhez tartozó Alsó-Bogát pusztán született 1905. március 27-én. Matematikai érdeklődése már középiskolás diák korában megmutatkozott, 1922-ben megnyerte az Eötvös Loránd Matematikai és Fizikai Társulat által az érettségizett diákok részére rendezett matematikai verseny első díját.

Érettségi után a Budapesti Tudományegyetem matematika—fizika szakos tanárjelölt hallgatója lett és hamarosan bekerült az Eötvös József Kollégiumba is.

Egyetemi tanulmányait 1927-ben befejezve a Budapesti Tudományegyetemen doktori szigorlatot tett; ezt követően a Vatea Elektroncsőgyárban dolgozott kutató fizikusként, majd még ugyanebben az évben tanársegédnek hívták meg a Szegedi Tudományegyetem Elméleti Fizikai Tanszékére, végül 1930-ban a Szegedi Tudományegyetem Bolyai Intézetéhez került, ahol Haar Alfréd és Riesz Frigyes adjunktusaként tevékenykedett.

Matematikai kutató tevékenysége igen széles területre terjedt ki. Foglalkozott az ún. nyílt játékok elméletével, majd ezt követően matematikai logikával. Matematikai logikai eredményei komoly nemzetközi elismerést váltottak ki; kiemelendők a matematikai logika eldöntésproblémájával kapcsolatos, a Gödel és a Church tételek egymáshoz való viszonyára vonatkozó, valamint az arithmetika axiomarendszerének konzisztenciájával kapcsolatos eredményei. A matematikai logikán kívül a matematika más ágaival is foglalkozott, jelentős, nemzetközi elismerést keltő eredményeket ért el az algebrában, a számelméletben, az analitikus számelméletben és az interpolációelméletben is. 1936-ban König Gyula jutalomban részesült. A második világháború utolsó legnehezebb éveinek elmúltával 1947-ben egyetemi tanárrá nevezték ki és megbízták a Bolyai Intézet Függvénytani Tanszékének vezetésével. 1949-ben a Magyar Tudományos Akadémia levelező tagja lett, majd 1950-ben a Kossuth díj III. fokozatával tüntették ki.

Már az ötvenedik életévén túl érdeklődése a matematikai logika különböző, elsősorban műszaki alkalmazásai felé irányult. Az ötvenes évek közepén kezdett hozzá a róla elnevezett logikai gép tervezéséhez, mely a későbbiekben azután megépítésre is került. Felismerve a számítógépek jövőbeni jelentőségét, szemináriumot szervezett az első Európában épült számítógép programozásának tanulmányozására. Ezen az úton kialakítva egy a számítástudományok oktatására alkalmas gárdát,

határozott kezdeményezésére 1957-ben Szegeden — az országban elsőként — beindulhatott a programozó matematikus szakképzés.

Kezdeti nehézségek után népes kutatógárda alakult ki Kalmár László professzor körül. Létrejött a József Attila Tudományegyetem Számítástudományi tanszéke, ezzel szoros egységben a Magyar Tudományos Akadémia Matematikai Logikai és Automataelméleti Kutató Csoportja, valamint a József Attila Tudományegyetem Kibernetikai Laboratóriuma; mindezek Kalmár László vezetésével. Jelentős és sok energiát lekötő oktatás- és tudományszervezési, oktatás- és tudománypolitikai tevékenysége mellett kimagasló eredményeket ért el a számítástudományban. Megalkotta a formulavezérlésű számítógép elvét, az ilyen típusú gép továbbfejlesztésén élete végéig aktívan dolgozott. Jelentősek matematikai nyelvészeti és információelméleti eredményei. 1961-ben a Magyar Tudományos Akadémia rendes tagjává választották, 1975-ben pedig az Állami díj első fokozatával tüntették ki. Ezek mellett 1958-ban Beke Manó Emlékdíjban részesült, 1964-ben az Oktatásügy Kiváló Dolgozója címmel, 1965-ben és 1970-ben pedig a Munka Érdemrend arany fokozatával tüntették ki, 1970-ben megkapta a Szele Tibor emlékérmet és 1975-ben a József Attila emlékérmet. Számos hazai és külföldi bizottságban tevékenykedett és több szakfolyóirat szerkesztésében vett részt.

1975-ben bekövetkezett nyugdíjbavonulása után is aktívan tevékenykedett. Tudományos kutató munkáját tovább folytatta, ugyanakkor továbbra is oktatott a József Attila Tudományegyetemen, tudományos irányításával rendszeresen segítette tanítványai szakmai előrehaladását, és végezte tudománypolitikai tevékenységét.

Kalmár László akadémikus igen széles látókörű matematikus volt. Kiemelendő az új iránti hallatlan érdeklődése és az a tulajdonsága, hogy nagyon határozott módon tudott küzdeni gondolatai megvalósítása érdekében. Egyaránt érdekelte és lelkesítette a matematika belső logikája, szépsége és alkalmazhatósága. Az a kutató egyéniség volt, aki át tudta hidalni a távolságot a matematikai alapkutatások és a matematika más tudományterületeken való alkalmazása között.

Élete egyik fő feladatának tekintette az oktató munkát és a tanítványaival való foglalkozást. A hazai matematikai logikusok közvetlenül vagy közvetve mindnyájan tanítványai. Az ország számítóközpontjaiban mindenütt találkozhatunk olyanokkal, akik az indíttatást Kalmár László akadémikustól kapták.

Emlékét kortársai és tanítványai ápolják, alkotásait pedig gondosan megőrzik.

Szerkesztőség

KALMÁR LÁSZLÓ MUNKÁSSÁGA

[1] Az interpolációról, *Mat. és Fiz. Lapok*, **33** (1927), 120—149. o.
[2] Zur Theorie der abstrakten Spiele, *Acta Sci. Math.*, **4** (1928), 65—85. o.
[3] Über die Abschätzung der Koeffizientensumme Dirichletscher Reihen, *Acta Sci. Math.*, **4** (1929), 155—181. o.
[4] Eine Bemerkung zur Entscheidungstheorie, *Acta Sci. Math.*, **4** (1929), 248—252. o.
[5] A "factorisatio numerorum" problémájáról, *Mat. és Fiz. Lapok*, **38** (1931), 1—15. o.
[6] Über die mittlere Anzahl der Produktdarstellungen der Zahlen, erste Mitteilung, *Acta Sci. Math.*, **5** (1931), 95—107. o.
[7] Ein Beitrag zum Entscheidungsproblem, *Acta Sci. Math.*, **5** (1932), 222—236. o.

[8] Ein Beweis des Ruffini—Abelschen Satzes, *Acta Sci. Math.*, **6** (1932), 59—60. o.
[9] Zum Entscheidungsproblem der mathematischen Logik, Verhandlungen des internationalen Mathematiker—Kongresses (Zürich, 1932), **2** kötet, 337—338. o.
[10] Über die Erfüllbarkeit derjenigen Zählausdrücke, welche in der Normalform zwei benachbarte Allzeichen enthalten, *Math. Annalen*, **108** (1933), 466—484. o.
[11] Über einen Löwenheimschen Satz, *Acta Sci. Math.*, **7** (1934), 112—121. o.
[12] Über die Axiomatisierbarkeit des Aussagenkalküls, *Acta Sci. Math.*, **7** (1935), 222—243. o.
[13] Zurückführung des Entscheidungsproblems auf den Fall von Formeln mit einer einzigen, binären, Funktionsvariablen, *Compositio math.*, **4** (1936), 137—144. o.
[14] A számelmélet alaptételéről, *Mat. és Fiz. Lapok*, **43** (1936), 27—45. o.
[15] Zur Reduktion des Entscheidungsproblems, *Norsk Mat. Tidsskrift*, **19** (1937), 121—130. o.
[16] Jelentés az 1938. évi König Gyula-jutalomról, *Mat. és Fiz. Lapok*, **45** (1938), 1—17. o.
[17] On the reduction of the decision problem, first paper: Ackermann prefix, a single binary predicate, *The journal of symbolic logic*, **4** (1939), 1—9. o.
[18] On the possibility of definition by recursion, *Acta Sci. Math.*, **9** (1940), 227—232. o.
[19] A Hilbert-féle bizonyításelmélet célkitűzései, módszerei és eredményei, *Mat. és Fiz. Lapok*, **48** (1941), 65—119. o.
[20] Egyszerű példa eldönthetetlen aritmetikai problémára, *Mat. és Fiz. Lapok*, **50** (1943), 1—23. o.
[21] On the reduction of the decision problem, second paper: Gödel prefix, a single binary predicate, *The journal of symbolic logic*, **12** (1947), 65—73. o. (Surányi Jánossal együtt.)
[22] Matematika és dialektikus materializmus. *Magyar Technika*, **3** (1948), 100—102. o.
[23] On unsolvable mathematical problems, Proceedings of the tenth International Congress of Philosophy (Amsterdam, 1948), I. kötet, 756—758. o.
[24] On the reduction of the decision problem, third paper: Pepis prefix, a single binary predicate, *The journal of symbolic logic*, **15** (1950), 161—173. o. (Surányi Jánossal együtt.)
[25] Eine einfache Konstruktion unentscheidbarer Sätze in formalen Systemen, *Methodos*, **2** (1950), 220—226. o.
[26] Une forme du théoreme de Gödel sous des hypotheses minimales, *Comptes rendus de l'Academie Paris*, **229** (1949), 963—965. o.
[27] Quelques formes générales du théoreme de Gödel, *Comptes rendus de l'Academie Paris*, **229** (1949), 1047—1049. o.
[28] Another proof of the Gödel—Rosser incompletability theorem, *Acta Sci. Math.*, **12** (1950), 38—43. o.
[29] On Cauchy's convergence test, *Hungarica Acta Math.*, **1** (1950), 109—112. o.
[30] Über die Cantorsche Theorie der reellen Zahlen, *Publicationes math.*, **1** (1950), 150—159. o.
[31] Contributions to the reduction theory of the decision problem, first paper: prefix $(x_1)(x_2)(Ex_3)\ldots(Ex_{n-1})(x_n)$, a single binary predicate, *Hungarica Acta Math.*, **2** (1950), 64—73. o.
[32] Contributions to the reduction theory of the decision problem, third paper: prefix $(x_1)(Ex_2)\ldots(Ex_{n-2})(x_{n-1})(x_n)$, a single binary predicate, *Hungarica Acta Math.*, **2** (1951), 19—38. o.
[33] Another proof of the Markov—Post theorem, *Hungarica Acta Math.*, **3** (1952), 1—27. o.
[34] Contributions to the reduction theory of the decision problem, fourth paper: reduction to the case of a finite set of individuals, *Hungarica Acta Math.*, **2** (1951), 125—143. o.
[35] Sur l'équation de translation, *Studia Math.*, **12** (1951), 112—116. o. (Aczél Jánossal és J. G. Mikusinskivel együtt).
[36] A matematika alapjaival kapcsolatos újabb eredmények, *MTA Mat. és Fiz. Oszt. Közl.*, **2** (1952), 89—103, 108—112. o.
[37] Az eldöntésprobléma visszavezetése logikai formulák véges halmazon való kielégíthetőségének kérdésére, Az I. Magyar Mat. Kongr. Kiadványai (1953), 163—190. o.
[38] A Bolyai—Lobacsevszkij-féle geometria hatása az axiomatikus módszer fejlődésére, *MTA Mat. és Fiz. Oszt. Közl.*, **3** (1953), 235—242. o.
[39] L'influence de la géometrie de Bolyai—Lobatchevski sur le developpement de la méthode axiomatique, *Hungarica Acta Math.*, **5** (1954), 117—126. o.
[40] K. Schröter egy, az általános rekurzív függvény fogalmának definíciójára vonatkozó problémájának megoldása, *MTA Mat. és Fiz. Oszt. Közl.*, **5** (1955), 103—127. o.
[41] Über ein Problem, betreffend die Definition les Begriffs der allgemein-rekursiven Funktion, *Zeitschrift für math. Logik und Grundlagen der Math.*, **1** (1955), 93—95. o.

[42] Közvetlen bizonyítás az eldöntésproblémának általános rekurzív algoritmussal való megoldhatatlanságára, *MTA Mat. és Fiz. Oszt. Közl.*, **6** (1956), 1—25. o.
[42*a*] Об одной гипотезе, применяемой в исследованиях о так называемых неразрешиных арифметических задачах. *Труды 3-го Всесоюзного Мат. С. езда* (Москва, 1956). **4** 227—237.
[43] Megjegyzés a halmazelmélet Gödel-féle axiómarendszeréhez, *Mat. Lapok*, **7** (1956), 26—42 és 218—229. o. (Hajnal Andrással együtt).
[44] An elementary combinatorial theorem with an application to axiomatic set theory, *Publicationse Math.*, **4** (1956), 431—449. o. (Hajnal Andrással együtt).
[45] Ein direkter Beweis für die allgemein-rekursive Unlösbarkeit des Entscheidungsproblem des Prädikatenkalküls der ersten Stufe mit Identität, *Zeitschrift für math. Logik und Grundlagen der Math.*, **2** (1956), 1—14. o.
[46] A matematikai logikáról, *Magyar Tudomány*, 1956, 369—391. o.
[47] Az ún. megoldhatatlan matematikai problémákra vonatkozó kutatások alapjául szolgáló Church-féle hipotézisről, *MTA Mat. és Fiz. Oszt. Közl.*, **7** (1957), 19—38. o.
[48] Über arithmetische Funktionen von unendlich vielen Variablen, welche an jeder Stelle bloss von einer endlichen Anzahl von Variablen abhängig sind, *Colloquium Mathematicum*, **5** (1957), 1—5. o.
[49] An argument against the plausibility of Church's thesis, *Constructivity in Mathematics*, 1957, 72—80. o.
[50] A new principle of construction of logical machines (Namur, 1958), Proc. of the 2nd International Congress of Cybernetics, 458—463. o.
[51] A practical infinitistic computer, Infinitistic Methods in the Foundations of Mathematics, Warsaw (1959), 347—362. o.
[52] Einige philosophische Probleme der Kybernetik, Naturwissenschaft und Philosophie, Beitrag zum Internationalen Symposium über Naturwissenschaft und Philosophie anlässlich der 550-Jahr-Feier der Karl-Marx-Universität, Leipzig, 1960, 389—401.
[52a] On a digital computer which can be programmed in a mathematical formula language. A II. Magyar Mat. Kongr. (előadáskivonatok), Budapest, 1960, V. 3—16.
[53] Über einen Rechenautomaten der eine mathematische Sprache versteht, *Zeitschrift für Angewandte Mathematik und Mechanik*, **40** (1960), T 64—65. o.
[54] Wissenschaftliche Abstraktion und die Anwendung mathematischer Methoden in Biologie und Medizin, Thesen und Referate zum Thema "Die philosophische Bedeutung der Anwendung der Kybernetik auf Biologie und Medizin" Arzt und Philosophie (Berlin, 1960), 132—133, 150, 164. o.
[55] A contribution to the translation of arithmetical operators (assignment statements) into the machine language of the M—3, 1961, (in Chinese).
[56] Algorithmische Sprachen und Programmierung von Rechenautomaten, Mathematische und physikalisch-technische Probleme der Kybernetik (Berlin, 1962), 147—176. o.
[57] Über eine Variante des Neumannschen selbstreproduzierenden Automaten, Mathematische und physikalisch-technische Probleme der Kybernetik (Berlin, 1962), 522—528. o.
[58] A kvalitatív információelmélet problémái, *MTA Mat. és Fiz. Oszt. Közleményei*, **12** (1962), 293—301. o.
[59] Über eine erkenntnistheoretische Wurzel des „Anti-Kybernetismus" Kybernetik in Wissenschaft, Technik und Wirtschaft der DDR (Berlin, 1962), 53—56. o.
[60] „Sejts és bizonyítsál!"?, *Magyar Tudomány*, 1963, 816—823. o.
[61] Matematikai és nyelvi structúrák, Általános Nyelvészeti Tanulmányok II. (1964), 11—74, 166—172, 295—304. o.
[62] On the problem of foundation of our knowledge, The Foundation of Statements and Decisions, (Warsaw, 1965), 13—19. o.
[62*a*] О вложении теории автоматических цифровых вычислительных машин в алгебраическую теорию автоматов Мура, Мили и Глушкова. Сборник «Теория конечных и вероятностных автоматов» (Москва, 1965). 93—99.
[63] Les calculatrices automatiques comme structures algéibrques, Prévisions, Calcul et Réalités (Les grandes problemes des sciences, N° 15), Paris, Gauthier-Villars, 1965, 9—22. o.
[64] Un modele algébrique de calculatrice automatique, Troisieme congres du calcul et de traitment de l'information de l'AFCALTI, Paris, 1965, 381—387. o.
[65] Foundations of Mathematics — Wither Now? Problems in the philosophy of mathematics, (Amsterdam 1967), 188—207. o.

[66] A matematikai nyelvészet eredményei a nyelvoktatásban. (Hozzászólásként elhangzott a TIT 1964-es Országos idegennyelvoktatási Konferenciáján.) Modern Nyelvoktatás 5 (1967) 1. 25—29. o.
[67] Meaning, synonymy and translation. *Computational linguistics* **6** (1967), 27—39. o.
[68] Le language comme structure algébrique. *Cahiers de Linguistique Thèoretique et appliquée* **4** (1967), 73—82. o.
[69] Pattern recognation and conditional reflexes (General Problems), 5th International Congress on Cybernetics, Namur, September 11—15th 1967.
[70] Znacsanije, szinonimija i perevod. Razrabotka masinnih (avtomaticseszkih) szisztem perevode sz odnovo jazüka na drugoj i ih primenenija. Bp. 1968. 374—390. o.
[71] On the problem of full utilization of the technical possibilities of computers in devising appropriate approximation methode for the solution of numerical problems. *Sulletin Mathématique de la Société des Sciences Mat. de la R. S. de Roumainée*, **12** (60), 1968. 1—5. o.
[72] A programozási nyelvekkel kapcsolatos további teendők. *Információ—Elektronika* **4** (1968) 251—254. o.
[73] Digitális számológépek és célgépek alkalmazása az orvosi diagnosztikában. *Orvos és Technika* **7** (1968) 14—18. o.
[74] Bevezetés a kibernetikába. *TIT Fizikai Kémiai Matematikai Szakosztályának Tájékoztatója* **15** (1969), 55—71. o.
[75] An intuitive representation of context-free languages. International Conference on Computational Linguistics (COLING), 1969, reprint no. 66. 1—10. o.
[76] A kibernetikáról, *Fizikai Szemle* **5** (1970), 129—134. o.
[77] Obál Ferenc—Kalmár László—Madarász István—Muszka Dániel—Such György, Cybernetic model of the regulation of the homeostatis of the organism, (abstract) First Join Congress of the Hungarian Societies of Biochemistry, Biophysics and Physiology, Pécs, October 12—14, 1967. Bp. 1967. 42. o.
[78] Ist ALGOL wirklich eine algorithmische Sprache? J. Dörr—G. Holtz, Tagungsbericht Automatentheorie und formale Sprachen, Oberwolfach, 1969. Mannheim, 190. 305—315. o.
[79] R. Péter's work in the theory of recursive functions. Les functions recursives ot leur applications Colloque International, Tihany, 1967. 1—11. o.
[80] „Fahnendiagramme" — ein anschauliches Hilfsmittel zur Angabe von Programmiersprachen. Tagungsbericht Formale Sprachen und Programmiersprachen. Oberwelfach, 1971.
[81] An algebraic model of systems of digital computers. International Symposium and Summer School on Mathematical Foundations of Computer Science, (Warsaw, 1972), 1—12. o.
[82] A számítástechnikai szakemberképzés problémái a tudományegyetemeken. *Felsőoktatási Szemle* **21** (1972) 9. 548—552. o.
[83] Beszélgetés a matematikáról, *A természet világa* **103** (1972), 8. 351—356. o.
[84] On a Measure of Divergence of a Context-free Language from Finite state Languages, Proc. Symposium on Algebraic linguistics held 10—12 February 1970. at Smolenica, Bratislava, 1973. Publishing House of the Slovak Academy of Sciences. 93—106. o. (Recuil linguistique de Bratislava, Vol. IV.).
[85] Az elektronikus digitális számítógépek eddigi fejlődése és a várható fejlődés fő irányai, 1972. 70 oldal (xerox)
[86] Belső gépi nyelvek, beleértve a magasszintű nyelveket, 1972. (140 oldal) (társszerzőkkel együtt), 1974. (xerox)
[87] A számítástechnikai szakemberképzés problémái. A számítástechnikai oktatás a hazai felsőoktatási intézményekben c. tudományos konferencián elhangzott előadások. Visegrád, 1974. máj. 13—14. Bp. 1974, Egyetemi Számítóközpont 25—30. o.
[88] Géptől független szemlélet kialakítása a programtervezés oktatásában. — A számítástechnikai oktatás a hazai felsőoktatási intézményekben c. tudományos konferencia, Visegrád, 1974. május 13—14. Előadások Bp. 1974. Főv. ny.
[89] A pedagógus a számítógépek korában. *Köznevelés*, **30** 1974. 20. 3—5. o.

Alkalmazott Matematikai Lapok 2 (1976) 157—170

MAXIMUM TRANZITÍV UTAK ÉS ALKALMAZÁSUK EGY GEOLÓGIAI PROBLÉMÁRA: RÉTEGTANI EGYSÉGEK LÉTREHOZÁSA[1]

KOVÁCS LÁSZLÓ BÉLA és DIENES ISTVÁN
Budapest

Egy geológiai probléma tanulmányozása új gráfelméleti problémához, egy irányított gráf ún. maximum tranzitív útjainak meghatározásához vezetett. Először a tranzitív utak néhány alapvető tulajdonságát mutatjuk be. Ezután részfeladatokat jelölünk ki és ezekre megoldó eljárásokat írunk le. Bizonyos, csúcspárokra vonatkozó mérőszámok bevezetése és vizsgálata is az alapprobléma megoldását segíti elő. A kiindulásul szolgáló geológiai problémát röviden vázoljuk és kitérünk a Dorogi medence adataival végzett kísérleti eredményekre is.

1. Bevezetés

A jelen dolgozat elsősorban egy adott irányított gráf maximum tranzitív útjainak meghatározásával foglalkozik. A tranzitív út az irányítatlan gráfokra definiált teljes algráf fogalmának egy lehetséges kiterjesztése irányított gráfokra. Ennek az új fogalomnak a bevezetésére geológiai kutatásainkkal kapcsolatban volt szükség, amelyet röviden a 2. szakaszban vázolunk. Ezután következik az alapprobléma felvetése, majd a tranzitív utak néhány fontos tulajdonsága.

Az 5.—7. szakaszok az alapfeladat megoldását elősegítő néhány részfeladatot írnak le és megoldásukra szolgáló eljárásokat vázolnak. Itt szerepel a maximum teljes részgráf keresés mint relaxáció, a csúcspárok kapcsolati száma és egy dekompozíciós eljárás. A 8. szakasz bemutat egy egzakt illetve egy heurisztikus eljárást, melynek segítségével elhagyhatók azok a csúcsok, amelyek nincsenek egyetlen optimális megoldásban sem, illetve a heurisztikus eljárás esetében nem valószínű, hogy benne vannak. A 9. szakasz az alapfeladat megoldására szolgáló, korlátozás és szétválasztás elvén működő, eljárás főbb tulajdonságait ismerteti. Végül a 10. szakasz a Dorogi medencével kapcsolatos jelenlegi vizsgálataink fő célkitűzéseit és eddigi kísérleti eredményeit vázolja.

Alapfeladatunk relaxációja, egy irányítatlan gráf maximum teljes részgráfjainak meghatározása, közvetlenül is megoldható [3, 4] valamint halmazlefedési feladatként is [2, 10, 12—16]. Megjegyzendő azonban, hogy a jelen dolgozatban vázolt sok gondolat, illetve eljárás jól felhasználható mind erre, mind más erősen struktúrált diszkrét feladat megoldására — így a teljes algráf problémára is, különösen, ha csak a legnagyobb elemszámú azaz maximum teljes részgráfokat keressük.

1) Ez a dolgozat a IX. Nemzetközi Matematikai Programozási Szimpoziumon (Budapest, 1976) elhangzott előadás irásbeli változata.

2. A geológiai probléma leírása

A kőzettestek térbeli kiterjedésének ismerete a geológiai problémák többségének megoldásához elengedhetetlen. Ilyenek például:

— az adott terület ásványvagyon mennyiségének meghatározása,
— a bányászatot befolyásoló vetők térbeli helyzetének közelítő meghatározása,
— a termelés irányítása.

A formális rétegtan módot nyújt ezen a területen bizonyos problémák egzakt megfogalmazására. A lehető legnagyobb számú rendezett kőzettest megkeresésének problémája fontos szerepet játszik a fent felsorolt és más földtani feladatok algoritmikus módon történő megoldásában.

A Föld (alkalmasan definiált) részét kőzettestnek nevezzük. A kőzettestek egy halmazát elsőrendű rétegtani egységnek hívjuk. Kőzettestek között számos térbeli rendezési relációt lehet definiálni. A T_i kőzettest a T_j felett van „32" értelemben, ha

$$P_{x,y}(T) \cap P_{x,y}(T_i) \cap P_{x,y}(T_j) \neq \emptyset \tag{2.1}$$

ahol

$$P_{x,y}(T) = \{(x, y) | \exists z\colon (x, y, z) \in T\}$$

és

$$\inf \{z | (x, y, z) \in T_i\} \geqq \sup \{z | (x, y, z) \in T_j\}$$

minden rögzített

$$(x, y) \in P_{x,y}(T) \cap P_{x,y}(T_i) \cap P_{x,y}(T_j)$$

számpárra. Ezt a precedencia relációt T_i „32" T_j-vel jelöljük. Ha a (2.1) összefüggés nem teljesül, akkor T_i és T_j nem összehasonlíthatók a T kutatási területen.

Tegyük fel, hogy definiáltuk geológiai testek egy $\{T_i\}_{i \in I}$ halmazát. Ekkor célunk, hogy találjunk ezen kőzettesteknek egy lehető legnagyobb rendezett részhalmazát, azaz egy $J \subset I$ indexhalmazt és a J halmazon megadott ϱ rendezést, amely rendelkezik a következő tulajdonsággal:
Tetszőleges $i \neq j$, $i, j \in J$ index párra az $i \varrho j$ relációból következik, hogy vagy T_i „32" T_j vagy T_i és T_j nem összehasonlíthatók a „32" reláció szerint (a T vizsgálati tartományon).

Megjegyzések.

1. A „32" helyett sok más reláció is használható.

2. Minthogy a vizsgálati tartomány csak a fúrások mentén ismert, a fent említett T tartományt az ismert fúrások egyesítésével közelíthetjük.

3. A probléma matematikai megfogalmazása

Adott egy $n \times n$ méretű **D** tiltási mátrix, amelynek elemei 0 vagy 1 értékűek és a főátlóban mindenütt 0 áll.

I. Feladat. Keresendő az alábbi feltételeknek eleget tevő leghosszabb

$$(j_1, j_2, \ldots, j_k)$$

rendezett index sorozat:

(3.1a) $$\{j_1, j_2, \ldots, j_k\} \subseteq \{1, 2, \ldots, n\}$$

(3.1b) $$j_r \neq j_s, \quad \text{ha} \quad r \neq s$$

(3.1c) $d_{j_r j_s} = 1$ minden r, s egész számpárra, melyre $1 \leqq r < s \leqq k$.

Megjegyzés. A tiltási mátrix a geológiai feladatból úgy nyerhető, hogy $d_{ij}=1$, ha az a feltevés, hogy az i-edik kőzettest megelőzi a j-edik kőzettestet, nem ellentmondó a fúrásokból kapott adatokkal, (azaz vagy minden fúrásban az i-edik kőzettest megelőzi a j-ediket, vagy nem fordulnak elő közös fúrásban) egyébként $d_{ij}=0$. A $d_{jj}=0$ feltevésnek nincs jelentősége, kizárólag az egyszerűbb felírási mód kedvéért tettük.

Az I. feladat megfogalmazható egy gráfelméleti problémaként is, ha egy új fogalmat vezetünk be. Szükségünk lesz az alábbi definíciókra:
Tekintsük a $G=(V, E)$ irányított gráfot, ahol
$V = \{v_1, v_2, \ldots, v_n\}$ a csúcsok halmaza és
$E = \{e_1, e_2, \ldots, e_m\}$ az irányított élek halmaza.
Amint az irodalomban szokásos, *elemi útnak* nevezzük a G gráf egymáshoz kapcsolódó éleinek egy

$$(v_{j_1}, v_{j_2}), (v_{j_2}, v_{j_3}), \ldots, (v_{j_{k-1}}, v_{j_k})$$

sorozatát, amely egyetlen csúcson sem halad át kétszer, azaz

$$v_{j_1}, \ldots, v_{j_k} \in V; \quad (v_{j_1}, v_{j_2}), \ldots, (v_{j_{k-1}}, v_{j_k}) \in E$$

és

$$j_s \neq j_s, \quad 1 \leqq r < s \leqq k.$$

Az elemi útban szereplő élek számát az út hosszának nevezzük. Minthogy tárgyalásunkban a csúcsok számának van jelentősége, az adódó összefüggések is így használhatók kényelmesebben, ezért bevezetjük a csúcsokban mért hosszat, amit *csúcsszámnak* fogunk nevezni.

Az elemi utat az általa érintett csúcsok rendezett halmazával fogunk megadni:

$$(v_{j_1}, v_{j_2}, \ldots, v_{j_k}).$$

3.1. DEFINÍCIÓ. A G irányított gráf

$$(v_{j_1}, v_{j_2}, \ldots, v_{j_k})$$

elemi útját *tranzitív út*nak nevezzük, ha

$$(v_{j_r}, v_{j_s}) \in E$$

minden egész r, s számpárra, melyre

$$1 \leqq r < s \leqq k.$$

3.2. DEFINÍCIÓ. A G irányított gráf egy

$$(v_{j_1}, v_{j_2}, \ldots, v_{j_k})$$

tranzitív útját *maximális tranzitív út*nak nevezzük, ha nem bővíthető, azaz a G

gráfban nem létezik a fenti k csúcsot tartalmazó tranzitív út, amelynek csúcsszáma $k+1$.

3.3. DEFINÍCIÓ. A G irányított gráf egy

$$(v_{j_1}, v_{j_2}, \ldots, v_{j_k})$$

tranzitív útját *maximum tranzitív út*nak nevezzük, ha a lehetséges legtöbb csúcsot tartalmazza, azaz a G gráfban nem létezik. olyan tranzitív út, amelynek csúcsszáma $k+1$. Ezek után a következő problémát vizsgáljuk:

II. Feladat. Adott egy

$$G = (V, E)$$

irányított gráf. Keresendő a G gráfban egy maximum tranzitív út.

A II. feladat ekvivalens az I. feladattal, ha a G gráfot a következőképpen definiáljuk:

$$V = \{v_1, v_2, \ldots, v_n\},$$
$$E = \{(v_j, v_k) | v_j, v_k \in V \text{ és } d_{jk} = 1\}. \tag{3.2}$$

A továbbiakban feltételezzük, hogy a G gráfot a (3.2) definiáló egyenlőségek segítségével nyertük a **D** mátrixból így a II. feladat megoldását tűzzük ki célul.

4. A tranzitív utak néhány alapvető tulajdonsága

Gyakran feltesszük, hogy csak olyan tranzitív utakat keresünk, amelyeknek csúcsszáma legalább L. Erre több okunk is lehet:

— L-nél rövidebbek nem érdekesek, vagy már ismertek korábbi számításokból, illetve az algoritmus előző fázisából.

— Abból a hipotézisből indulunk ki, hogy létezik a G gráfban olyan tranzitív út, amelynek csúcsszáma legalább L. (lásd a 6.1. lemmát.)

Az első három lemma a tranzitív utak néhány egyszerű tulajdonságát mutatja, melyek jól felhasználhatók algoritmikus meghatározásuknál.

4.1. LEMMA. Ha teljesül a

$$\sum_{i=1}^{n} (d_{ij} + d_{ji} - d_{ij} d_{ji}) \leqq L-2 \tag{4.1}$$

egyenlőség, akkor a v_j csúcsot a G gráfnak egyetlen olyan tranzitív útja sem tartalmazhatja, melynek csúcsszáma legalább L.

Bizonyítás. A

$$d_{ij} + d_{ji} - d_{ij} d_{ji}$$

kifejezés akkor és csak akkor 1, ha a d_{ij} és d_{ji} számok közül legalább az egyik 1, azaz a v_i és v_j csúcsok állhatnak ugyanabban a tranzitív útban. Másszóval a (4.1) egyenlőtlenség baloldala azoknak a csúcsoknak a száma, amelyek a v_j csúccsal állhatnak közös tranzitív útban, így a lemmát igazoltuk.

Ez a lemma iteratív módon használható a csúcsok törlésére, ugyanis miután ezen kritérium segítségével néhány csúcsot kihagytunk lehetséges, hogy az egyenlőtlenség a megmaradó algráfban újabb csúcsokra teljesül.

A következő állítások kimondásához minden csúcshoz néhány jelzőszámot definiálunk és számítunk ki:

$$b_j = \sum_{i=1}^{n} d_{ij} \tag{4.2}$$

$$a_j = \sum_{k=1}^{n} d_{jk} \tag{4.3}$$

$$h_j = \sum_{i=1}^{n} (d_{ij} + d_{ji} - d_{ij} d_{ji}). \tag{4.4}$$

Ezek jelentése a következő: azoknak a csúcsoknak a száma, amelyek a v_j csúccsal közös tranzitív útban állhatnak a v_j csúcs előtt (b_j) után (a_j) vagy bárhol az útban (h_j). Megjegyzendő, hogy

$$h_j = a_j + b_j - \sum_{i=1}^{n} d_{ij} d_{ji}. \tag{4.5}$$

Ezt az egyenlőséget beláthatjuk akár közvetlenül a (4.2)—(4.4) összefüggésekből, akár az a_j, b_j, h_j számok jelentéséből, figyelembe véve, hogy $d_{ij} d_{ji} = 1$ akkor és csak akkor, ha v_i a v_j csúcs előtt és után egyaránt állhat egy tranzitív útban.

Jelöljük a II. feladat optimumát, azaz a G gráf egy maximum tranzitív útjának csúcsszámát $L_{\max}$-szal.

Nyilvánvalóan

$$L_{\max} \leqq 1 + \max_j h_j \tag{4.6}$$

azonban ennél sokkal jobb korlátot is kaphatunk az alábbi állítás segítségével:

4.2. LEMMA. Legyenek a G gráf csúcsai úgy indexelve, hogy a

$$h_1 \geqq h_2 \geqq \ldots \geqq h_n$$

egyenlőtlenségek teljesüljenek. Vezessük be a

$$h'_j = \min (j-1, h_j), \quad j = 1, 2, \ldots, n$$

jelölést és tegyük fel, hogy $h_2 = 1$. Ekkor az alábbi egyenlőtlenség teljesül:

$$L_{\max} \leqq 1 + \max_j h'_j. \tag{4.7}$$

Bizonyítás. Jelölje

$$h'_k = \max_j h'_j.$$

Tegyük fel indirekt módon, hogy

$$L_{\max} \geqq 2 + h'_k.$$

Ez az egyenlőtlenség azt jelenti, hogy létezik $2 + h'_k$ csúcsot tartalmazó tranzitív út, azaz van $2 + h'_k$ számú olyan v_j csúcs, amely mindegyike legalább $1 + h'_k$ csúccsal

állhat közös útban. Másszóval

$$h_j \geqq 1 + h'_k \geqq k$$

legalább

$$2 + h'_k \geqq 1 + k$$

számú csúcsra. A csúcsok rendezéséből következik, hogy

$$h_{k+1} \geqq k,$$

ami azt jelenti, hogy

$$h_{k+1} \geqq k > \min(k-1, h_k) = h_k.$$

Ez az egyenlőtlenség sorozat cáfolja az indirekt feltevésünket, amivel a lemmát bebizonyítottuk.

A következő lemma szükséges feltételt ad olyan tranzitív út létezésére, amelynek csúcsszáma legalább L.

4.3. LEMMA. Ha létezik a G gráfban tranzitív út, melynek csúcsszáma L, akkor van L csúcs, amelyekhez tartozó

$$(b_{j_k}, a_{j_k}), \quad k = 1, 2, \ldots, L$$

számláló párokra teljesülnek a

$$\begin{aligned} b_{j_k} &\geqq k-1, \quad k = 1, 2, \ldots, L, \\ a_{j_k} &\geqq L-k, \quad k = 1, 2, \ldots, L \end{aligned} \tag{4.8}$$

egyenlőségek.

Bizonyítás. Tekintsünk egy tetszőleges tranzitív utat, melynek csúcsszáma L. Ebben az útban a k-adik csúcsot $k-1$ másik csúcs előzi meg és $L-k$ követi, így ezekre a csúcsokra teljesülni kell a (4.8) egyenlőtlenségeknek.

Megjegyzés. Általában elég hosszadalmas a fenti feltételek ellenőrzése egy-egy megadott L értékre. Ehelyett gyengébb, de könnyebben ellenőrizhető feltételeket is találhatunk, például:
Tetszőleges k egész értékre $(0 \leqq k \leqq L)$ képezzük az alábbi halmazokat:

$$\{v_j | v_j \in V,\ b_j \geqq k\} \quad \text{és} \quad \{v_j | v_j \in V,\ a_j \geqq L-k\}. \tag{4.9}$$

Ahhoz, hogy létezzék tranzitív út, melynek csúcsszáma L, szükséges, hogy a (4.9) halmazok legalább $L-k$ illetve k számú elemet tartalmazzanak.

A következő lemma segítségével $L_{\max}$ értékére egyre jobb felső korlátot kaphatunk. Tekintsük a V csúcshalmaznak egy tetszőleges S részhalmazát, és tegyük fel, hogy valamilyen eljárással meg tudunk határozni az S halmazbeli maximum tranzitív út csúcsszámára vonatkozó $L(S)$ felső korlátot. A tényleges maximumot jelölje $L_{\max}(S)$. Vezessük be a következő jelölést:

$$S_k = \{v_j | v_j \in V,\ h_j \leqq k-1\}, \tag{4.10}$$

ahol a h_j számokat a (4.5) egyenlőség definiálta.

4.4. LEMMA. Tetszőleges pozitív egész k értékre, melyre

(4.11) $$k < L(S)$$

és

(4.12) $$L(S-S_k) < L(S),$$

teljesül az alábbi egyenlőtlenség.

(4.13) $$\hat{L}(S) = \max\{k, L(S-S_k)\} \geqq L_{\max}(S).$$

Bizonyítás. Tegyük fel (feltételes korlátozásként — lásd a bizonyítás utáni jegyzetet), hogy

(4.14) $$L_{\max}(S) \geqq k+1.$$

Függetlenül attól, hogy a (4.14) egyenlőtlenség igaz-e vagy sem, bizonyos következtetéseket mindenképpen levonhatunk. Ha (4.14) teljesül, akkor mindazok a csúcsok, amelyek nem állhatnak közös tranzitív útban legalább k másik csúccsal az optimum megváltoztatása nélkül elhagyhatók, azaz

(4.15) $$L_{\max}(S) \leqq L(S-S_k).$$

Másrészt, ha a (4.14) feltétel hamis, akkor

(4.16) $$L_{\max}(S) \leqq k.$$

Mindkét esetben a (4.13) felső korlát korrekt, és így az eredeti, $L(S)$ felső korlátnál határozottan kisebbet kaptunk a (4.11) és (4.12) egyenlőtlenségek szerint.

Nyilvánvaló, hogy ezt a lemmát egymást követően többször is alkalmazhatjuk egy alkalmas stratégiának megfelelően. Akár magasabb, akár alacsonyabb k értékkel is indulhatunk, vagy felezési eljárást is használhatunk. A k értékek megfelelő választásában az S_k halmazoknak is van szerepük.

Megjegyzés: A fenti lemma lényege a feltételes korlát bevezetése, amely más nagyméretű struktúrált diszkrét programozási feladat megoldásában is fontos szerepet játszhat. Ennek a korlátozásnak az a szerepe, hogy két részre bontja a megengedett megoldások halmazát és mindkettőre az eredetinél jobban kezelhető, a célfüggvény értékére jobb korlátokat kapnak. A feltételes korlátozás nevét BALAS adta [1] előadásában. Ezt az elvet egymástól függetlenül fedeztük fel és adtuk elő a IX. Matematikai Programozási Szimpoziumon. BALAS általánosabb feladatra fogalmazta meg elvét azonban a miénktől teljesen eltérő módon. A gondolat szoros kapcsolatban van az általánosított korlátozás és szétválasztási elvvel is.

5. Relaxáció

A következő módon a II. feladat egy jó relaxációjához juthatunk. Tekintsük a

$$\bar{G} = (\overset{\bullet}{V}, \bar{E})$$

irányítatlan gráfot, amelyet a 2. szakaszban definiált irányított gráfból olymódon nyerhetünk, hogy az irányításoktól eltekintünk, azaz

$$\bar{E} = \{(v_i, v_j) | (v_i, v_j) \in E \text{ és/vagy } (v_j, v_i) \in E\}.$$

A $\bar{G}$ gráfban a (v_i, v_j) és (v_j, v_i) éleket azonosnak tekintjük, más szóval a $\bar{G}$ gráfban sincsenek többszörös élek. A

$$\bar{G}_N = (N, \bar{E}_N)$$

gráfot a $\bar{G}=(V, E)$ gráf *maximális teljes algráf*jának nevezzük, ha teljesül az alábbi három feltétel:

(i) $\bar{G}$ a $\bar{G}$ algráfja, azaz

$$N \subseteq V, \quad \bar{E}_N = \{(v_i, v_j) | v_i, v_j \in N, (v_i, v_j) \in \bar{E}\}$$

(ii) $\bar{G}_N$ teljes gráf:
minden $v_i, v_j \in N$ esetén $(v_i, v_j) \in \bar{E}_N$

(iii) $\bar{G}_N$ maximális teljes algráf:
bármely $N \subset M \subset V$ esetén a $\bar{G}_M$ algráf nem teljes.

A $\bar{G}_N$ maximális teljes algráfot *maximum teljes*nek nevezzük, ha az

$$M \subseteq V \quad \text{és} \quad |M| > |N|$$

relációkból következik, hogy $\bar{G}_M$ nem teljes, más szóval $\bar{G}_N$ a lehetséges legnagyobb számú elemet tartalmazó teljes algráf. A következő lemma a relaxáció lényegét fejezi ki:

5.1. LEMMA. Ha a $\bar{G}=(V, \bar{E})$ irányítatlan gráf egy maximum teljes algráfja $\bar{G}_N=(N, \bar{E}_N)$ olyan tulajdonságú, hogy a megfelelő irányított algráf $G_N=(N, E_N)$ nem tartalmaz körutat, akkor az

$$N = \{v_{k_1}, v_{j_2}, \ldots, v_{j_k}\}$$

csúcshalmaz rendezhető oly módon, hogy a $G=(V, E)$ irányított gráf maximum tranzitív útját, azaz II. feladat optimális megoldását kapjuk.

Bizonyítás. Nyilvánvalóan, amennyiben

$$(v_{r_1}, v_{r_2}, \ldots, v_{r_t})$$

tranzitív út a G gráfban, akkor a $\bar{G}_M=(M, \bar{E}_M)$ a $\bar{G}$ gráf teljes részgráfja, ahol

$$M = \{v_{r_1}, v_{r_2}, \ldots, v_{r_t}\}.$$

Ezért a lemma bizonyításához elegendő megadni egy tranzitív utat, amely az N halmaz pontjaiból áll. Minthogy a G_N gráfban nincs körút, létezik legalább egy csúcs, amelyhez nincs negatív él. (v_j csúcshoz negatív él: $(v_i, v_j) \in E_N$ pozitív él: $(v_j, v_k) \in E_N$.) Jelöljük ezt a csúcsot v_{r_1}-gyel. Az eljárás megismételhető az $N - v_{r_1}$ csúcshalmazra, stb. Az eredményül kapott út

$$(v_{r_1}, v_{r_2}, \ldots, v_{r_k})$$

tranzitív út, mert a fenti eljárás szerint nincsen

$$(v_{r_q}, v_{r_p}) \in E_N \quad (p < q)$$

típusú él. Minthogy a $\bar{G}_N$ gráf teljes, tehát

$$(v_{r_p}, v_{r_q}) \in E_N \quad \text{minden} \quad 1 \leqq p < q \leqq k \quad \text{esetén},$$

ami igazolja a fenti út tranzitivitását és így a lemmát.

Algoritmikus szempontból ezt az eredményt többféle módon fel lehet használni. Ezek közül a legegyszerűbb, hogy amennyiben a lemma feltételei nem teljesülnek, akkor a körutakat egyes csúcsok elhagyásával megszüntetjük. Az így adódó tranzitív utak nem feltétlenül optimálisak, azonban a II. feladatnak megengedett megoldását adják és a későbbiekben vázolt algoritmusban fontos szerepet játszhatnak.

Megjegyzés. Ha a $G_N=(N, E_N)$ gráf valamennyi körútját meg tudjuk szüntetni az élek egy $F\subset E_N$ halmazának elhagyásával, olymódon, hogy a

$$\bar{G}'_N = (N, \overline{E_N - F}) \tag{5.1}$$

továbbra is teljes gráf, akkor a lemma állítása igaz marad. Ezért a csúcsok elhagyása előtt élek elhagyását kell megkísérelni, mert így $|N|$ nem csökken.

A következő szakaszban egy mérőszámot vezetünk be abból a célból, hogy egyes csúcspárok együttes előfordulásának „jóságát" mérjük.

6. Csúcspárok kapcsolati száma

Két különböző csúcs együttesen annál jobb számunkra, minél több olyan csúcsa van, amely mindkettővel állhat együtt egy tranzitív útban. Ezért az alábbi halmazok segítségével bevezetjük majd a v_i, v_j csúcspár ún. kapcsolati számát:

$$S(v_i, v_j) = \{v_k | v_k \in V,\ d_{ik}+d_{ki} \geqq 1,\ d_{jk}+d_{kj} \geqq 1\} \tag{6.1}$$

$$S_L(v_i, v_j) = \{v_k | v_k \in S(v_i, v_j) \text{ és } h_k \geqq L-1\}, \tag{6.2}$$

ahol L jelentése: csak olyan tranzitív utakat keresünk, amelyeknek csúcsszáma legalább L. Ezek után a v_i, v_j csúcspárra vonatkozó L-től függő *kapcsolati szám*ot a következőképpen definiáljuk:

$$f_{ij}(L) = \begin{cases} |S_L(v_i, v_j)|, & \text{ha } d_{ij}+d_{ji} \geqq 1, \\ 0, & \text{egyébként.} \end{cases} \tag{6.3}$$

A következő lemma további iteratív szűrő eszközt szolgáltat a II. feladat megoldásához például a csúcsok számának csökkentéséhez.

6.1. LEMMA. Ha az

$$R_k = \{v_i | v_i \in V,\ f_{ik}(L) \geqq L-2\} \tag{6.4}$$

halmaz $L-1$-nél kevesebb számú elemet tartalmaz, akkor egyetlen tranzitív út sem létezik, amelynek csúcsszáma L (vagy nagyobb) és tartalmazza a v_k csúcsot.

Bizonyítás. Ha a lemma feltétele teljesül, akkor legfeljebb $L-2$ számú olyan v_i csúcs létezik, amelyre a következő állítások teljesülnek:

1° Létezhet olyan tranzitív út, amelynek csúcsszáma L és tartalmazza mind a v_i mind a v_k csúcsot.

2° Legalább $L-2$ olyan csúcs van, amely mind a v_i mind a v_k csúccsal állhat együtt tranzitív útban.

Az 1° és 2° feltételek szükségesek olyan tranzitív út létezéséhez, amelynek csúcsszáma L és mind a v_k mind a v_i csúcsot tartalmazza. Minthogy az ilyen v_i csúcsok száma legfeljebb $L-2$, ez a v_k csúccsal együtt is legfeljebb $L-1$, amivel a lemma állítását beláttuk.

7. Nagy feladatok dekompozíciója

Ha a II. feladat mérete nagy, azaz a V halmaz néhány száz vagy még több elemet tartalmaz, akkor rendkívül nehéz maximum tranzitív utat találni. Ebben az esetben megkísérelhetjük a feladatot dekomponálni. Egy lehetséges eljárást az alábbiakban vázolunk.

Dekompozíciós eljárás

1. *Lépés.* Dekomkonáljuk a $G=(V, E)$ irányított gráfot $G_k=(N_k, E_k)$ algráfokra felhasználva az előző szakaszban leírt kapcsolati számokat. A felbontásnál

$$V = \bigcup_{k=1}^{r} N_k, \quad N_i \cap N_j = \emptyset, \quad i \neq j$$

és G_k a G gráfból az N_k csúcshalmaz által indukált algráf. A következő két típusú N_k halmazt részesítjük előnyben:

a) $L_{\max}(N_k) \ll |N_k|$ ($L_{\max}(N_k)=1$ vagy 2, ha lehetséges)
b) $L_{\max}(N_k)$ $|N_k|$-hoz képest elég nagy.
$L_{\max}(N_k)$ a G_k gráfbeli maximum tranzitív út csúcsszámát jelenti.

2. *Lépés.* Határozzuk meg a $\bar{G}_k$, $k=1, \ldots, r$ irányítatlan gráfokban egy-egy maximum teljes algráfot, melynek elemszámát jelöljük n_k-val.

3. *Lépés.* Határozzuk meg az egyes részproblémák kapcsolatát. Jelöljön $m_j(i, k)$, $k=1, 2, \ldots, n_i$ egy jó felső korlátot arra vonatkozóan, hogy az N_j halmazból legfeljebb hány elemet választhatunk ki, ha N_i-ből k számút már kiválasztottunk, úgy hogy együttesen teljes algráf csúcsait adhassák.

4. *Lépés.* Alkalmazzunk egy leszámlálási eljárást, amelyben az x_k diszkrét változó az N_k halmazból kiválasztott elemek számát jelöli.

Ez az eljárás együttesen alkalmazható a korábban leírt szűrési módszerrel, amint azt a következő szakasz második részében láthatjuk.

8. Szűrés

Ezt a kifejezést a jelen dolgozatban olyan csúcsok elhagyási módszerére használjuk, amelyek a II. feladat egyetlen optimális megoldásában sem lehetnek benne illetve nem valószínű, hogy benne vannak attól függően, hogy az egzakt-vagy a heurisztikus változatról beszélünk. Nyilvánvalóan az itt következő eljárások a dolgozat első 6 szakasza lemmáinak az eredményeivel is kombinálhatók.

Szűrés a kapcsolati számok segítségével — heurisztikus eszköz

Fontos, hogy feladatunknak minél előbb, minél jobb megoldásait kapjuk, mert ennek segítségével egyrészt csúcsok elhagyása válik lehetővé, másrészt az optimumra vonatkozó jó alsó illetve felső korlátokat nyerhetünk, ami az algoritmus további menetét döntően befolyásolhatja. Ismét feltéve, hogy csak azokkal a tranzitív utakkal foglalkozunk, amelyeknek csúcsszáma legalább L, az alábbiakban egy egyszerű heurisztikus eljárást vázolunk:

1. *Lépés.* Válasszunk egy „jó csúcs"-ot, amelyet jelöljünk v_{j_1}-gyel. (Például választhatjuk a legnagyobb kapcsolati számú csúcsok egyikét.) Legyen $r=1$.

2. *Lépés.* Töröljünk minden v_k csúcsot, melyre

$$f_{j_r,k}(L) < L-2$$

és ismételjük ezt a lépést a törlések után újra számított kapcsolati számokkal, mindaddig, amíg további elemeket elhagyhatunk.

3. *Lépés.* Válasszunk egy további jó csúcsot a megmaradtak közül $v_{j_{r+1}}$ például az utoljára kapott kapcsolati számok segítségével. Növeljük meg r értékét eggyel.

Ismételjük a 2. és 3. lépéseket felváltva, amíg valamennyi csúcsot töröltük. Ekkor a

$$(v_{j_1}, v_{j_2}, \ldots, v_{j_r})$$

csúcshalmaz a $\bar{G}$ irányítatlan gráf maximális (de nem feltétlenül maximum) teljes algráfját határozza meg, amelyből a II. feladat egy megengedett megoldásához juthatunk az irányított körök megszüntetésével (lásd a 4. szakaszt.)

Egzakt szűrés a kapcsolati számok segítségével a dekompozíciós eljárásban

A 6. szakaszban leírt eljárást javíthatjuk szűrés alkalmazásával. A 6.1. lemma állítását erősebbé tehetjük a következő módon (az ottani jelöléseket használva):

8.1. LEMMA. Ha

$$t_p = \sum_{k=1}^{r} \min \{n_k, |R_p \cap N_k|\} < L-1, \tag{8.1}$$

akkor egyetlen tranzitív út sem létezik, melynek csúcsszáma L és tartalmazza a v_p csúcsot.

Bizonyítás. Elegendő megmutatni, hogy azoknak a csúcsoknak a száma, amelyek a v_p csúccsal együtt előfordulhatnak, egy L csúcsszámú tranzitív útban, legfeljebb t_p. A 6.1. lemmában beláttuk, hogy $|R_p|$ ezen pontok számának felső korlátja. Az R_p halmazt felbonthatjuk a következőképpen:

$$R_p = \bigcup_{k=1}^{r} (R_p \cap N_k).$$

Azonban az N_k halmaznak legfeljebb n_k eleme állhat közös tranzitív útban az n_k

számok definíciójának megfelelően (l. a 6. szakasz eljárásának 2. pontját), amivel a bizonyítást befejeztük.

Ez az eredmény még tovább javítható az N_k csúcshalmazok között a 3. lépésben definiált kapcsolati számok segítségével.

9. Egy korlátozás és szétválasztási algoritmus vázlata

A II. feladat megoldására szolgáló algoritmus lényege az $L_{\max}$ alsó és felső korlátjának (L_w ill. L_u) fokozatos javítása, amíg a kettő egybeesik. Alsó korlátokat a megengedett megoldások szolgáltatnak, felső korlát számítására pedig több módszert is leírtunk. Az alábbiakban röviden vázoljuk a megoldásul szolgáló korlátozás és szétválasztás típusú keret algoritmus néhány fontos tulajdonságát:

1° Alsó korlátokat és jó megengedett megoldásokat kaphatunk a 8. szakaszban leírt módon.

2° Felső korlátokat a 4.1.—4.3., 6.1 és 8.1. lemmák segítségével iteratív szűrő eljárásokkal kaphatunk.

3° Mindkét korlátot javíthatjuk, ha felváltva számítjuk azokat és felhasználjuk a 4.4. lemma feltételes megszorítási módszerét. Így a felső és alsó korlát közötti felezési eljárás is lehetségessé válik.

4° Főképpen először mélységben haladó eljárást kell alkalmazni, hogy mielőbb megengedett megoldáshoz jussunk.

5° A 4. szakaszban leírt relaxációs eljárást használva további jó megoldásokat nyerhetünk.

6° További heurisztikus részeljárásokkal gyorsíthatjuk az algoritmus menetét, például jó csúcs részhalmazok választhatók a 6. szakaszban leírt módon, amelyből nyert tranzitív utakat egyes csúcsok időszakos elhagyása után megkísérelhetünk bővíteni.

7° Nagyobb feladatok megoldásához a 6. szakaszban leírt és más dekompozíciós eljárások, a 8. szakaszbeli egzakt szűrési módszer és általában több módszer hatékony kombinálása szükséges.

10. Kísérletek a Dorogi medence adataival

A barnaszén bányászat első évszázada során erről a területről igen sok adat vált ismertté. A terület rétegtanát legutóbb GIDAI [9] foglalta össze.

A Dorogi medencével kapcsolatos kutatásaink fő céljai:

1° Olyan módszert találjunk, amely alkalmas földtani, kőzettani és teleptani térképek, valamint szelvények számítógépi megrajzolásához szükséges segédadatok előállítására.

2° A vágathajtás operatív irányításához segédeszköz biztosítása.

3° Egy — a felhasznált adatok alapján előállítható — lehető legrészletesebb formális kronosztratigráfiai skála és kronosztratigráfiai beosztás felállítása.

Az MTA CDC 3300 számítógépén létrehoztunk egy adatfile-t, amely a Dorogi medence 47 fúrás eocénnek leírt szakaszában meghatározott teljes fauna adatait tartalmazza. Több mint 1000 ősmaradvány taxon fúrásonkénti első és utolsó elő-

fordulása segítségével definiáltuk azon kőzettesteket, amelyeket a további számításokhoz használtunk.

Az összes adatok száma meghaladta a 20 000-t.

Eddig csak kísérleti eredmények állnak rendelkezésre. A leggyakoribb 150 kőzettestet felhasználva, az egész területre vonatkozó 8 rendezett kőzettestet találtunk. Másik 8 kőzettestet találtunk egy részterületre szigorúbb feltételek mellett. Egy másik 252 elemű kőzettest halmazból, melyet a fentitől eltérő módon definiáltunk — 26 rendezett kőzettestet nyertünk.

Az eredmények hasznosaknak mutatkoztak az 1° feladat megoldásában (l. DIENES és KOVÁCS [7, 8]. További kutatás tárgya a 2° és 3° feladat, valamint nagyobb kiinduló kőzettest halmazból az 1° feladatra is nagyobb rendezett kőzettest halmazt kívánunk meghatározni.

IRODALOM

[1] BALAS, E., *Set Covering With Cutting Planes from Conditional Bounds*, presented on IX. International Symposium on Mathematical Programming, 23—27 August, 1976. Printed as Management Science Research Report No. 399. Carnegie—Mellon University Pittsburgh, Pennsylvania 15 213

[2] BALAS E. and PADBERG, M. V., "On the set overing roblem", *Operations Research* **20** (1972) 1152—1161.

[3] BRON, C. and KERBOSCH, J., "Algorithm 457-Finding all cliques of an undirected graph", *Comm of ACM* **16** (1973) 575—577.

[4] CHRISTOFIDES, N. *Graph Theory — An Algoritmic Approach* (Academic Press, 1975).

[5] DIENES, I., "Subdivision of a geological body into ordered parts", *Matematika és Számítástechnika a nyersanyagkutatásban* (MFT. kiadv. Szerk: Dienes I.) c. kötetben, 1974.

[6] DIENES, I. (MANN, C. J. ed. coll.) *Formalized Theoretical Stratigraphy: A Formalization of Strarigraphic Terminology*. (J. IAMG, 1977).

[7] DIENES, I. and KOVÁCS, L. B. "Formalized eocene stratigraphy of the Dorog Basin, Transdanubia, Hungary, *Acta Geologica* (in prep.)

[8] DIENES, I. and KOVÁCS, L. B. "An algorithm for setting up optimal chronostratigraphic scales and plotting haemera tables", *Computers and geosciences* (in prep.)

[9] GIDAI, L. «L'Eocéne de la region de Dorog Ann. Inst.» *Geol. Publ. Hung.* **LV** 1973.

[10] KOVÁCS, L. B. "A new solution for the set covering probem, *Proceedings of the 5 th IFIP Conference on Optimization Techniques*, Rome 1973, Lecture Notes in Computer Science 4—5, Springer, 1974.

[11] KOVÁCS, L. B. A computer program system for the solution of pure linear discrete problems", in: *Colloquia Mathematica Societatis János Bolyai 12. Progress in Operations Research, Eger, (Hungary)* , Ed. A. Prékopa, (North Holland, 1975) 573—589.

[12] KOVÁCS, L. B. *Discrete Programming*, North Holland (to appear)

[13] LEMKE, C. E., SALKIN, H. M. and SPIELBERG, K., "Set covering by single branch enumeration with linear programming subproblems", *Operations Research* **19** (1971) 998—1022.

[14] MARSTEN, R. E. "An algorithm for large set partitioning problems", *Management Science* **20** (1974) 779—787.

[15] SALKIN, H. M. and KOCAL, R. D., "Set covering by an all integer algorithm: Computational experience", *ACM Journal* **20** (1973) 189—193.

[16] THIRIEZ, H. M., "The set covering problem: A group theoretic approach", *RAIRO* **5** (1971) 83—104.

(Beérkezett: 1977. augusztus 3.)

KOVÁCS LÁSZLÓ BÉLA
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1111 BUDAPEST, KENDE U. 13—15

DIENES ISTVÁN
MAGYAR FÖLDTANI INTÉZET
1143 BUDAPEST, NÉPSTADION ÚT 14.

MAXIMUM TRANSITIVE PATHS AND THEIR APPLICATION TO A GEOLOGICAL PROBLEM: SETTING UP STRATIGRAPHIC UNITS

L. B. KOVÁCS and I. DIENES

The study of a geological problem has led to a new graph theoretic problem, the determination of the so called maximum transitive path, which is a counterpart of the complete subgraph problem in undirected graphs. Some basic properties of these transitive paths are discussed. Then, as large sized problems are to be solved, further subproblems, procedures and measurements are introduced and examined to promote the solution of the basic problem. The geological problem, the main goals and some experiment with the Dorog basin is also described shortly.

Alkalmazott Matematikai Lapok 2 (1976) 171—187

A STABIL SZTOCHASZTIKUS PROGRAMOZÁSI MODELLRŐL

MAYER JÁNOS
Budapest

A dolgozatban a *Prékopa-féle* STABIL *sztochasztikus programozási modell*el foglalkozunk nemlineáris programozási szempontból. A redukált gradiens módszerből kiindulva szerkesztünk egy algoritmust a modell megoldására. A konvergencia bizonyítása után a módszerrel kapcsolatos numerikus tapasztalatainkat ismertetjük.

1. Bevezetés

Ebben a dolgozatban a STABIL *sztochasztikus programozási modell*hez kapcsolódó nemlineáris programozási problémák numerikus megoldásával foglalkozunk. A modell konstrukciója PRÉKOPA ANDRÁSTÓL származik, a STABIL elnevezés a [12] dolgozatban került bevezetésre. PRÉKOPA [8], [9], [10] eredményeire alapozva lehetővé válik a konvex programozás algoritmusainak alkalmazása a STABIL modell esetében. PRÉKOPA, DEÁK, GANCZER, PATYI [12] illetve DEÁK [2] dolgozataikban ZOUTENDIJK egy megengedett irányos módszere, az ún. P2 módszer [18] sikeres alkalmazásáról számolnak be. SZÁNTAI [16] egy támaszsík módszeren alapuló algoritmust alkalmazott a modell megoldására és igen jó számolási eredményeket közöl. RAPCSÁK [14] a SUMT módszer különböző változatait alkalmazta a modell megoldására. Jelen munka célja egy, a *redukált gradiens módszer*en [1], [17] alapuló további megoldási lehetőség bemutatása.

A dolgozat első részében a STABIL valószínűséggel korlátozott sztochasztikus programozási modellt ismertetjük a nemlineáris programozási szempontból fontos tulajdonságokat kiemelve. A következő rész a nemlineáris programozás egyik leghatékonyabb algoritmusát, a redukált gradiens módszert ismerteti, majd a STABIL modellre való alkalmazás numerikus tapasztalatait foglaljuk össze. Ezután az általunk javasolt algoritmust ismertetjük és bebizonyítjuk az eljárás konvergenciáját. Végül beszámolunk a módszerrel kapcsolatos számítógépes tapasztalatainkról.

E helyen szeretnék köszönetet mondani PRÉKOPA ANDRÁSnak az értékes tanácsokért, melyeket munkám során tőle kaptam.

2. A STABIL sztochasztikus programozási modell

Tekintsük a következő, PRÉKOPA ANDRÁSTÓL származó sztochasztikus programozási modellt:

$$\begin{aligned} &\max f(\mathbf{x}), \\ &G(\mathbf{x}) \geqq p, \\ &\mathbf{Ax} = \mathbf{b}, \\ &\mathbf{x} \geqq \mathbf{0}, \end{aligned} \tag{2.1}$$

ahol $\mathbf{x} \in R^n$, $\mathbf{b} \in R^m$, $\mathbf{A}$ $m \times n$-es mátrix, $m < n$, $r(\mathbf{A}) = m$, azaz $\mathbf{A}$ teljes rangú. Tételezzük fel, hogy $f(\mathbf{x})$ az R^n-en konkáv, folytonosan differenciálható függvény, és a nemlineáris feltételi függvény a következő alakú: $G(\mathbf{x}) = P\{\mathbf{Sx} \geqq \boldsymbol{\beta}\}$, ahol $\mathbf{S}$ $r \times n$-es mátrix, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_r)^T$ valószínűségi vektorváltozó. Így $G(\mathbf{x})$ annak a valószínűsége, hogy az $\mathbf{Sx} \geqq \boldsymbol{\beta}$ véletlen jobboldali vektorral megadott egyenlőtlenségrendszert $\mathbf{x}$ kielégíti, a feltételben szereplő p pedig előírt valószínűségi szint.

Nemlineáris programozási szempontból (2.1) speciális típusú, csak egy nemlineáris feltételt tartalmazó feladat. A modellhez kapcsolódó nemlineáris programozási apparátus PRÉKOPA következő alapvető tételére épül: [8], [9], [10].

2.1. TÉTEL: Ha $\beta_1, \ldots, \beta_r$ együttes eloszlása folytonos és az együttes sűrűségfüggvény $e^{-Q(\mathbf{z})}$ alakú, $\mathbf{z} \in R^r$ ahol $Q(\mathbf{z})$ az egész R^r téren értelmezett konvex függvény, akkor $G(\mathbf{x})$ logaritmikusan konkáv R^n-en.

A dolgozat további részében feltesszük, hogy $\beta_1, \ldots, \beta_r$ együttes eloszlása nem elfajult normális eloszlás. Ekkor érvényes:

2.2. LEMMA: A fenti feltételek teljesülése esetén:

a) $G(\mathbf{x})$ logaritmikusan konkáv függvény,

b) $\nabla G(\mathbf{x})$ *Lipschitz-folytonos*, azaz van olyan $L > 0$ konstans, hogy minden $\mathbf{x} \in R^n$, $\mathbf{y} \in R^n$-re

$$\|\nabla G(\mathbf{x}) - \nabla G(\mathbf{y})\| \leqq L \cdot \|\mathbf{x} - \mathbf{y}\|,$$

c) ha létezik olyan megengedett $\mathbf{y}$, melyre $G(\mathbf{y}) > p$ *(Slater-feltétel)*, akkor a (2.1) feladat esetében a *Kuhn—Tucker feltételek* az optimalitás szükséges és elegendő feltételei.

Bizonyítás: Az a) állítás a 2.1. tétel közvetlen következménye, b) könnyen adódik a többdimenziós normális eloszlásfüggvény tulajdonságait kihasználva. [12]-ben PRÉKOPA bebizonyította, hogy a *Slater-feltétel* teljesülése esetén a *Kuhn—Tucker feltételminősítés* a megengedett tartomány minden pontjában érvényes, amiből c) adódik.

3. A redukált gradiens módszer

Ebben a részben a nemlineáris programozás egyik leghatékonyabb algoritmusával, a redukált gradiens módszerrel [1], [17] foglalkozunk és megmutatjuk, hogyan alkalmazható az algoritmus a STABIL modell megoldására. A módszert a nemlineáris programozás általános feladatára adjuk meg, majd megmutatjuk, hogyan specializálódik az eljárás a (2.1) feladat esetében.

A nemlineáris programozás alapfeladatát a következő alakban tekintjük:

$$\begin{aligned} &\max f(\mathbf{x}) \\ &g_i(\mathbf{x}) = 0, \quad i = 1, \ldots, m, \\ &\mathbf{x} \geqq \mathbf{0} \quad \mathbf{x} \in R^n, \quad m < n, \end{aligned} \tag{3.1}$$

ahol $f(\mathbf{x})$, $g_i(\mathbf{x})$, $i=1, \ldots, m$ az egész téren értelmezett folytonosan differenciálható függvények. Jelöljük D-vel a megengedett megoldások halmazát. Legyen $\mathbf{x}^{(1)}$ az eljárás egy megengedett indulópontja, feltesszük, hogy megadható hozzá olyan $I_1 \subset \{1, \ldots, n\}$ indexhalmaz, melyre a $\frac{d\mathbf{g}(\mathbf{x}^{(1)})}{d\mathbf{x}}$ *Jacobi-mátrix* I_1-be tartozó indexű oszlopai bázist alkotnak, és $x_i^{(1)} > 0$ minden $i \in I_1$-re. Itt $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_m(\mathbf{x}))^T$.

Tételezzük fel, hogy $\nabla g_1(\mathbf{x}), \ldots, \nabla g_m(\mathbf{x})$ lineárisan függetlenek az $\{\mathbf{x} | f(\mathbf{x}) \geqq f(\mathbf{x}^{(1)})\} \cap D$ halmazon.

Az algoritmust indukcióval adjuk meg, feltesszük, hogy $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}$-t már kiszámoltuk, és $\mathbf{x}^{(k)}$-hoz van olyan bázisa a $\frac{d\mathbf{g}(\mathbf{x}^{(k)})}{d\mathbf{x}}$ *Jacobi-mátrix*nak, hogy ha I_k jelöli a bázisindexek halmazát, akkor $x_i^{(k)} > 0$ minden $i \in I_k$-ra. A módszer áttekinthetőbb tárgyalása érdekében feltesszük, hogy az első m oszlop alkotja a bázist, és vektorainkat particionáljuk a következő módon:

$$\mathbf{x} = (\mathbf{y}, \mathbf{z}), \quad \mathbf{w} = (\mathbf{u}, \mathbf{v}), \quad \mathbf{y} \in R^m, \quad \mathbf{u} \in R^m, \text{ ahol } \mathbf{w} \text{ a lépés iránya.}$$

Ekkor $\mathbf{x}^{(k+1)}$-et a következő lépéssorozattal határozzuk meg:

1. *lépés.* Az irány meghatározása.

Invertáljuk $\frac{d\mathbf{g}(\mathbf{x}^{(k)})}{d\mathbf{y}}$-t, majd kiszámoljuk a következő vektort:

$$\mathbf{r}^T = \nabla_{\mathbf{z}}^T f(\mathbf{x}^{(k)}) - \nabla_{\mathbf{y}}^T f(\mathbf{x}^{(k)}) \left(\frac{d\mathbf{g}(\mathbf{x}^{(k)})}{d\mathbf{y}} \right)^{-1} \frac{d\mathbf{g}(\mathbf{x}^{(k)})}{d\mathbf{z}}. \tag{3.2}$$

Ezután megoldunk egy iránykereső segédfeladatot:

$$\begin{aligned} &\max \mathbf{r}^T \mathbf{v} \\ &v_i \geqq 0, \quad \text{ha} \quad z_i^{(k)} = 0, \quad i = 1, \ldots, n-m, \\ &\|\mathbf{v}\| \leqq 1. \end{aligned} \tag{3.3}$$

Legyen (3.3) optimális megoldása $\mathbf{v}^{(k)}$. Ha az optimális célfüggvényérték 0, az eljárás véget ér, $\mathbf{x}^{(k)}$ a (3.1) feladat *Kuhn—Tucker pont*ja. Különben:

$$\mathbf{u}^{(k)} = -\left(\frac{d\mathbf{g}(\mathbf{x}^{(k)})}{d\mathbf{y}} \right)^{-1} \frac{d\mathbf{g}(\mathbf{x}^{(k)})}{d\mathbf{z}} \cdot \mathbf{v}^{(k)}, \quad \mathbf{w}^{(k)} = (\mathbf{u}^{(k)}, \mathbf{v}^{(k)}). \tag{3.4}$$

2. *lépés.* Maximalizáljuk $f(\mathbf{x})$-et az $\mathbf{x}^{(k)} + \lambda \mathbf{w}^{(k)}$, $\lambda \geqq 0$ félegyenes és a pozitív ortáns metszetén, legyen $\hat{\lambda}$ az optimumhoz tartozó λ érték. Ha $\hat{\lambda} = +\infty$ adódna, $\hat{\lambda} = M$-et veszünk, ahol M elég nagy, az eljárás során rögzített szám.

3. *lépés.* Visszatérés a megengedett felületre.
Vezessük be a következő jelöléseket:

$$\hat{\mathbf{y}} = \mathbf{y}^{(k)} + \hat{\lambda}\mathbf{u}^{(k)}, \quad \hat{\mathbf{z}} = \mathbf{z}^{(k)} + \hat{\lambda}\mathbf{v}^{(k)}.$$

Világos, hogy $(\hat{\mathbf{y}}, \hat{\mathbf{z}})$ általában nem megengedett megoldás, ezért alkalmas módon vetíteni kell a megengedett felületre.

Alkalmazzuk *Newton módszerét* a következő egyenletrendszer megoldására:

$$g_i(\mathbf{y}, \mathbf{z}^{(k)} + \hat{\lambda}\mathbf{v}^{(k)}) = 0, \quad i = 1, \ldots, m, \tag{3.5}$$

indulópontként $\mathbf{y} = \hat{\mathbf{y}}$-ot véve.

Amennyiben az eljárás során:

a) valamelyik $y_i \leqq 0$ lesz,

b) a célfüggvény fogy,

c) a *Newton-módszer* lassan konvergál, akkor $\hat{\lambda}$-ot helyettesítjük $\frac{1}{2}\hat{\lambda}$-val és kiszámolva a megfelelő $\hat{\mathbf{y}}, \hat{\mathbf{z}}$-ot újra indítjuk a *Newton-módszer*t az új $\hat{\mathbf{y}}$ indulópontból. Amennyiben az újraindítás az a) eset miatt következett be, egyúttal tárolunk egy olyan bázisindexet, amelyik a bajt okozta. Az eljárás eredményeként adódik (3.5) egy $\tilde{\mathbf{y}}$ közelítő megoldása. A legutolsó $\hat{\lambda}$-hoz tartozó $\hat{\mathbf{z}}$-ot véve:

$$\mathbf{y}^{(k+1)} = \tilde{\mathbf{y}}, \quad \mathbf{z}^{(k+1)} = \hat{\mathbf{z}}.$$

4. *lépés.* Ha a 3. lépésben fellépett az a) eset, megváltoztatjuk a bázisindexek halmazát, a kritikus index helyett másikat választunk. Az eljárás ezután $\mathbf{x}^{(k+1)}$-ből indulva az 1. lépéssel folytatódik.

Ezzel a redukált gradiens módszer vázát megadtuk. Látható, hogy a leírásban számos hézag található, ezeket konkrét tartalommal megtöltve adódnak a módszer különböző variánsai. Mivel jelen dolgozatban célunk csupán az algoritmus alapgondolatának érzékeltetése, ezért további részletek kidolgozása helyett röviden diszkutáljuk az eljárást.

Az 1. lépésben meghatározott irány a következő segédfeladat megoldása:

$$\max\,[\nabla_{\mathbf{y}}^T f(\mathbf{x}^{(k)})\mathbf{u} + \nabla_{\mathbf{z}}^T f(\mathbf{x}^{(k)})\mathbf{v}],$$

$$\frac{d\mathbf{g}(\mathbf{x}^{(k)})}{d\mathbf{y}}\mathbf{u} + \frac{d\mathbf{g}(\mathbf{x}^{(k)})}{d\mathbf{z}}\mathbf{v} = \mathbf{0}, \tag{3.6}$$

$$v_i \geqq 0, \quad \text{ha} \quad z_i^{(k)} = 0, \quad i = 1, \ldots, n-m,$$

$$\|\mathbf{v}\| \leqq 1,$$

azaz az $\mathbf{x}^{(k)}$ pontból induló $\mathbf{w} = (\mathbf{u}, \mathbf{v})$ irányú félegyenes az e pontban a $g_i(\mathbf{x}) = 0$, $_i = 1, \ldots_m$ felületekhez fektetett érintőhipersíkok metszetében halad, lokálisan nem vezet ki a pozitív ortánsból és a célfüggvény $\mathbf{w}$ irányú iránymenti deriváltja maximális. Ha most, mint feltettük $\frac{d\mathbf{g}(\mathbf{x}^{(k)})}{d\mathbf{y}}$ nem szinguláris, akkor (3.6)-ban az egyenletrendszer-rész eliminálható, és adódik a (3.3) iránykereső feladat. Az elimináció során kapott $\mathbf{r}$ vektor a módszer nevében szereplő redukált gradiens.

A (3.3) segédfeladat megoldása egyszerű, ha az euklidesi normát használjuk, akkor a megoldás explicite megadható:

$$(3.7)\qquad \tilde{v}_i^{(k)} = \begin{cases} 0, & \text{ha } z_i^{(k)} = 0 \text{ és } r_i < 0, \\ r_i, & \text{különben.} \end{cases} \quad \text{és } \mathbf{v}^{(k)} = \frac{1}{\|\tilde{\mathbf{v}}^{(k)}\|}\tilde{\mathbf{v}}^{(k)}$$

Ha a $\|\mathbf{v}\| = \sum_{i=1}^{n-m} |v_i|$ normát választjuk, akkor a kapott irány a szimplex módszer iránya, feltéve persze, hogy (3.1) lineáris programozási feladat és $\mathbf{x}^{(k)}$ megengedett bázismegoldás.

3.1. LEMMA. Ha a (3.3) feladat optimális célfüggvényértéke 0, akkor $\mathbf{x}^{(k)}$ a (3.1) *Kuhn—Tucker pont*ja. Ellenkező esetben $f(\mathbf{x})$ az $\mathbf{x}^{(k)}+\lambda\mathbf{w}^{(k)}$, $\lambda \geqq 0$ félegyenes mentén $\mathbf{x}^{(k)}$ környezetében lokálisan növelhető.

Bizonyítás. Ha (3.3)-ban az optimális célfüggvényérték 0, akkor 0 lesz az optimális célfüggvényérték abban a feladatban is, amit (3.6)-ból a normálási feltétel elhagyásával kapunk.

Alkalmazható *Farkas tétele*, és a *Kuhn—Tucker feltételek* adódnak. Ha pedig (3.3)-ban az optimális célfüggvényérték pozitív, akkor az $f(\mathbf{x})$ függvény $\mathbf{w}^{(k)}$ irányú iránymeneti deriváltja az $\mathbf{x}^{(k)}$ pontban pozitív, ebből a második állítás következik.

Az algoritmus kritikus lépése a 3. lépés. Megmutatjuk, hogy ha $\hat{\lambda}$ elég kicsi, akkor a (3.5) nemlineáris egyenletrendszer megoldható:

3.2. LEMMA. Ha $\hat{\lambda}$ elég kicsi, akkor a (3.5) egyenletrendszernek $\mathbf{y}^{(k)}$ valamely környezetében fix $\hat{\lambda}$ mellett létezik olyan egyértelmű $\tilde{\mathbf{y}}$ megoldása, melyre $f(\tilde{\mathbf{y}}, \mathbf{z}^{(k)}+\hat{\lambda}\mathbf{v}^{(k)}) > f(\mathbf{y}^{(k)}, \mathbf{z}^{(k)})$.

Bizonyítás. Mivel $\dfrac{d\mathbf{g}(\mathbf{x}^{(k)})}{d\mathbf{y}}$ nem szinguláris, alkalmazhatjuk az implicit függvényekre vonatkozó tételt. E szerint elég kicsi λ-ra egyértelműen definiálva van egy olyan $\mathbf{y}(\lambda)$ függvény, melyre $\mathbf{y}(0)=\mathbf{y}^{(k)}$ és $\mathbf{g}(\mathbf{y}(\lambda), \mathbf{z}^{(k)}+\lambda\mathbf{v}^{(k)})=0$ teljesül, azaz elég kicsi λ-ra az $(\mathbf{y}(\lambda), \mathbf{z}^{(k)}+\lambda\mathbf{v}^{(k)})$ pont a felületen fekszik. Ha most a célfüggvényt ezen a görbén tekintjük, azaz vesszük az $F(\lambda)=f(\mathbf{y}(\lambda), \mathbf{z}^{(k)}+\lambda\mathbf{w}^{(k)})$ függvényt, akkor az implicit függvények deriváltjára vonatkozó tételt alkalmazva $F'(0)=\|\mathbf{v}^{(k)}\|^2>0$, azaz a célfüggvény a görbén az $\mathbf{x}^{(k)}$ pont környezetében lokálisan nő. Ezzel a lemmát bebizonyítottuk.

A redukált gradiens módszer lineáris feltételekre WOLFE-tól származik [17], majd ABADIE és CARPENTIER fejlesztette tovább nemlineáris feltételekre [1].

A lineáris feltételek esetére FAURE és HUARD publikált egy hibás konvergenciabizonyítást [4], majd kiderült, hogy a módszer ebben az alakjában nem feltétlenül konvergens. A konvergencia biztosítására többen továbbfejlesztették az eljárást, így HUARD maga is. Szép konvergenciabizonyítást közöl egy módosított változatra KLEINMICHEL [6].

A továbbiakban megvizsgáljuk, hogy hogyan specializálható a redukált gradiens módszer a (2.1) STABIL modell esetére. Az egyszerűbb jelölésmód kedvéért feltesszük, hogy a bázist az első $m+1$ oszlop alkotja, és bevezetjük a következő jelöléseket:

$$\mathbf{x} = (\bar{\mathbf{y}}, y_{m+1}, \mathbf{z}), \quad \mathbf{A} = (\mathbf{B}, \mathbf{a}_{m+1}, \mathbf{C})$$

ahol $\bar{\mathbf{y}} \in R^m$, $\mathbf{B}$ $m \times m$-es mátrix.

Ekkor a bázis a következő:

$$\bar{\mathbf{B}} = \begin{pmatrix} \dfrac{dG(\mathbf{x}^{(k)})}{d\bar{\mathbf{y}}} & \dfrac{\partial G(\mathbf{x}^{(k)})}{\partial y_{m+1}} \\ \mathbf{B} & \mathbf{a}_{m+1} \end{pmatrix}. \tag{3.8}$$

Ha most $\mathbf{B}$ nem szinguláris és $\mathbf{B}^{-1}$ ismeretes, a kikövített bázis inverze is könnyen kiszámolható, ha

$$t = \frac{\partial G(\mathbf{x}^{(k)})}{\partial y_{m+1}} - \frac{dG(\mathbf{x}^{(k)})}{d\bar{\mathbf{y}}} \mathbf{B}^{-1}\mathbf{a}_{m+1} \neq 0.$$

Ha az inverz bázist a (3.8)-nak megfelelő módon blokkokra bontjuk: $\bar{\mathbf{B}}^{-1} = \begin{pmatrix} \mathbf{c}^T & d \\ \mathbf{P} & \mathbf{q} \end{pmatrix}$, $\mathbf{c} \in R^m$, $\mathbf{q} \in R^m$, $d \in R^1$, $\mathbf{P}$ $m \times m$-es mátrix, akkor könnyen adódik:

$$d = \frac{1}{t}; \quad \mathbf{q} = -d\mathbf{B}^{-1}\mathbf{a}_{m+1}; \quad \mathbf{c}^T = -t\frac{dG(\mathbf{x}^{(k)})}{d\bar{\mathbf{y}}}\mathbf{B}^{-1}; \quad \mathbf{P} = \mathbf{B}^{-1}(\mathbf{E}_m - \mathbf{a}_{m+1} \cdot \mathbf{c}^T)$$

ahol $\mathbf{E}_m$ egy $m \times m$-es egységmátrix.

Feltesszük, hogy induló bázisunk fenti szerkezetű, és az iterációk során ezt a bázisszerkezetet igyekszünk megőrizni. Így elegendő $\mathbf{B}^{-1}$-et tárolni és a bázis-cseréknél transzformálni. Ennek a kezelésmódnak egy további előnye is van, ugyanis az algoritmus kritikus 3. lépése a következő alakot ölti: Megoldandó a következő egyenletrendszer:

$$\begin{aligned} \mathbf{B}\bar{\mathbf{y}} + \mathbf{a}_{m+1}y_{m+1} &= \mathbf{b} - \mathbf{C}\hat{\mathbf{z}}, \\ G(\bar{\mathbf{y}}, y_{m+1}, \hat{\mathbf{z}}) &= p, \quad \text{ha} \quad G(\mathbf{x}^{(k)}) = p. \end{aligned} \tag{3.9}$$

A rendszer lineáris részének, azaz az első m sornak a megoldáspontjai egy egyenes mentén helyezkednek el, ami $\mathbf{B}^{-1}$ segítségével pl. y_{m+1}-gyel parametrizálva megadható. A feladat eme egyenes és a nemlineáris felület metszéspontjának meghatározása.

4. Számítógépes experimentáció a redukált gradiens módszerrel

Ebben a részben összefoglaljuk a (2.1) STABIL modell megoldására irányuló számítógépes kísérleteinket, amelyek végül az 5. rész algoritmusához vezettek.

A STABIL modell megoldására irányuló első kísérletként elkészült a redukált gradiens módszer számítógépes programja a (3.1) általános nemlineáris probléma esetére. Ez a 3. részben ismertetett algoritmusvázra épült, a hézagok egy részét kísérletileg töltve be. A felhasznált standard eszközök közül néhányat az alábbiakban kiemelünk.

Az algoritmus gyorsítására két báziscsere között a *Fletcher—Reeves módszer*t használtuk [13]. Az eljárás konjugált gradiensekkel dolgozik, és bár konvergenciasebessége nem vetekszik a *kvázi-Newton módszerek*kével, kis memóriaigénye miatt ezt választottuk, és tapasztalataink szerint jól bevált, különösen nagyobb méretű feladatoknál.

A mátrixinvertálás nagyon időigényes művelet, ezért az optimumpont közelében felhasználtuk az inverz mátrix egy approximációját. Ez, a matematikai programozás irodalmában *Beale-approximáció*nak nevezett közelítés a következő egyszerű lineáris algebrai észrevételen alapul: Ha valamely $\mathbf{A}$ mátrix $\mathbf{A}^{-1}$ inverze adott és $\mathbf{A} \sim \mathbf{B}$, azaz $\mathbf{A}$ keveset változik, akkor meg lehet próbálkozni $\mathbf{B}^{-1}$ következő közelítő kiszámolásával:

$$\mathbf{B}^{-1} \sim (2\mathbf{E}-\mathbf{A}^{-1}\mathbf{B})\mathbf{A}^{-1}, \quad \text{ahol } \mathbf{E} \text{ egységmátrix.} \tag{4.1}$$

Ez az approximáció egy, az inverz mátrix értékét pontosító iterációs eljárás első lépése. Maga az eljárás a következő. Legyen $\mathbf{B}^{-1}$ egy közelítése $\mathbf{B}_0$, akkor a

$$\mathbf{B}_{k+1} = (2\mathbf{E}-\mathbf{B}_k\mathbf{B})\mathbf{B}_k \tag{4.2}$$

rekurzív összefüggéssel definiált $\mathbf{B}_k$ mátrixsorozat $\mathbf{B}^{-1}$-hez tart bizonyos feltételek teljesülése esetén. (4.2)-ből látható, hogy

$$\mathbf{E}-\mathbf{B}_k\mathbf{B} = (\mathbf{E}-\mathbf{B}_0\mathbf{B})^{2k}, \tag{4.3}$$

azaz a konvergencia feltétele $(\mathbf{E}-\mathbf{B}_0\mathbf{B})^{2k} \to 0$ $(k \to \infty)$. Ez pl. teljesül, ha $\mathbf{E}-\mathbf{B}_0\mathbf{B}$ maximális abszolút értékű sajátértéke <1.

A redukált gradiens módszer 2. lépése egyenes menti maximalizálás, erre a célra az aranymetszési arányon alapuló eljárás jól bevált, az optimumot a félegyenes mentén nem számoltuk ki nagy pontossággal.

Az algoritmus számítógépes megvalósítása során a legtöbb gondot a 3. lépés jelentette. A gyakorlatban $\hat{\lambda}$ egyszerű felezése nem volt kielégítő, bonyolultabb stratégiát kellett választani. GERENCSÉR [5] szellemes eljárást javasol a 3. lépés kiküszöbölésére.

Az elkészült programot egy sor determinisztikus feladaton teszteltük, ezek közül a legnagyobb 17 nemlineáris feltételt tartalmazott, a változók száma 37. A számítógépes eredményekről [7]-ban beszámoltunk.

Ezek után megkíséreltük egy (2.1) típusú 2 változós feladat megoldását a programmal, az eredmény: teljes kudarc. Hogy ennek okát megérthessük, foglaljuk össze a (2.1) típusú feladatok numerikus megoldása során felmerülő nehézségeket. Két fő nehézség adódik:

a) $G(\mathbf{x})$ értékének kiszámolásához az r-változós normális eloszlásfüggvény értékeit kell meghatározni, azaz integrálni kell a r-dimenziós térben. DEÁK [2], [3] vizsgálataiból kiderült, hogy erre a célra szimulációs technikák a legalkalmasabbak.

b) A (2.1) problémában szereplő nemlineáris feltétel feltételi függvényét, $G(\mathbf{x})$-et csak közelítőleg tudjuk kiszámolni, és a számolás viszonylag nagy gépidő-igényű. Célunk csupán a b) sajátosság hatásainak a vizsgálata.

Visszatérve a redukált gradiens módszerre: az algoritmus minden esetben az iteráció 3. lépésénél akadt el, nem volt képes visszatérni a megengedett felületre. Ennek oka az, hogy $G(\mathbf{x})$ gradiense is csak közelítőleg számolható ki, a hibák a *Jacobi-mátrix* invertálása során torzultak, és a *Newton módszer*rel ezért nem sikerült a (3.5) egyenletrendszert megoldani.

Következő kísérletként a redukált gradiens módszer (2.1) feladatra specializált változatával dolgoztunk. Az eljárást a 3. részben vázoltuk. Itt a kritikus 3. lépés a következő feladatra redukálódott: Meghatározandó egy egyenes és a $G(\mathbf{x})=p$

felület metszéspontja, és az egyenest a lineáris feltételek adják. Ez egy stabil eljárás, sikerrel alkalmaztuk (2.1) típusú kisméretű feladatok megoldására. Mivel azonban $G(\mathbf{x})$-et csak közelítőleg lehet kiszámolni, az egyenes és a felület metszéspontjának meghatározása túl sok $G(\mathbf{x})$-kiszámítást igényel, és az eljáráshoz kapcsolódó mátrix-manipulációk is elég időigényesek, ezért az algoritmussal kapott számolási idők nem voltak kielégítőek.

Ezek a negatív tapasztalatok arra a gondolatra vezettek bennünket, hogy teljesen ki kellene küszöbölni a kritikus 3. lépést a redukált gradiens módszernél. Az új algoritmus csak megengedett pontokkal dolgozik, de megőrzi azt a hatékony technikát, amivel a redukált gradiens módszer a lineáris feltételeket kezeli.

A továbbiakban a (2.1) feladat jelöléseit használjuk. Tételezzük fel, hogy adott valamely $\mathbf{x}\in R^n$ pont, és a következő partíció: $\mathbf{x}=(\mathbf{y}, \mathbf{z})$, $\mathbf{y}\in R^m$ az $\mathbf{A}$ mátrix hasonlóan van particionálva: $\mathbf{A}=(\mathbf{B}, \mathbf{C})$, ahol $\mathbf{B}_{m\times m}$ típusú és nem szinguláris, továbbá $\mathbf{y}>\mathbf{0}$. Legyen a $\mathbf{w}$ irányvektor hasonlóképp particionálva:

$$\mathbf{w} = (\mathbf{u}, \mathbf{v}), \quad \mathbf{u}\in R^m.$$

Legyen $\vartheta>0$ rögzitett szám.

Ekkor ZOUTENDIJK P1 módszerének iránykereső feladata: [18]

$$\begin{aligned} &\max \zeta \\ &\nabla_{\mathbf{y}}^T f(\mathbf{x})\mathbf{u}+\nabla_{\mathbf{z}}^T f(\mathbf{x})\mathbf{v} \geqq \zeta, \\ &\nabla_{\mathbf{y}}^T G(\mathbf{x})\mathbf{u}+\nabla_{\mathbf{z}}^T G(\mathbf{x})\mathbf{v} \geqq \vartheta\zeta, \quad \text{ha} \quad G(\mathbf{x}) = p, \\ &\mathbf{B}\mathbf{u}+\mathbf{C}\mathbf{v} = \mathbf{0}, \\ &v_i \geqq 0, \quad \text{ha} \quad z_i = 0, \quad i = 1, \ldots, n-m, \\ &\|(\mathbf{u}, \mathbf{v})\| \leqq 1. \end{aligned} \tag{4.4}$$

A redukált gradiens módszer megfelelő iránykereső feladata: [1]

$$\begin{aligned} &\max \zeta \\ &\nabla_{\mathbf{y}}^T f(\mathbf{x})\mathbf{u}+\nabla_{\mathbf{z}}^T f(\mathbf{x})\mathbf{v} \geqq \zeta, \\ &\nabla_{\mathbf{y}}^T G(\mathbf{x})\mathbf{u}+\nabla_{\mathbf{z}}^T G(\mathbf{x})\mathbf{v} \geqq 0, \quad \text{ha} \quad G(\mathbf{x}) = p, \\ &\mathbf{B}\mathbf{u}+\mathbf{C}\mathbf{v} = \mathbf{0}, \\ &v_i \geqq 0, \quad \text{ha} \quad z_i = 0, \quad i = 1, \ldots, n-m, \\ &\|\mathbf{v}\| \leqq 1. \end{aligned} \tag{4.5}$$

A javasolt módszer a következő iránykereső feladatokkal dolgozik:

$$\begin{aligned} &\max \zeta \\ &\nabla_{\mathbf{y}}^T f(\mathbf{x})\mathbf{u}+\nabla_{\mathbf{z}}^T f(\mathbf{x})\mathbf{v} \geqq \zeta, \\ &\nabla_{\mathbf{y}}^T G(\mathbf{x})\mathbf{u}+\nabla_{\mathbf{z}}^T G(\mathbf{x})\mathbf{v} \geqq \vartheta\zeta, \quad \text{ha} \quad G(\mathbf{x}) = p, \\ &\mathbf{B}\mathbf{u}+\mathbf{C}\mathbf{v} = \mathbf{0}, \\ &v_i \geqq 0, \quad \text{ha} \quad z_i = 0, \quad i = 1, \ldots, n-m, \\ &\|\mathbf{v}\| \leqq 1. \end{aligned} \tag{4.6}$$

A (4.6) iránykereső feladat lehetővé tesz egy, a redukált gradiens módszerhez hasonló redukciót, ugyanakkor megengedett irányt generál. A lineáris feltételek kezelése $\mathbf{B}^{-1}$ tárolásával és a báziscseréknek megfelelő transzformációjával történik. A (4.6) iránykereső feladatra épülő algoritmus nem lenne konvergens, a konvergencia biztosítására ún. ε_k-aktív feltételekkel dolgozunk, ennek részleteit az 5. részben ismertetjük.

Az ebben a részben említett programok FORTRAN nyelven íródtak az MTA CDC 3300-as gépére.

5. Algoritmus a STABIL modell megoldására

Ebben a részben egy új algoritmust ismertetünk a (2.1) probléma megoldására. Az algoritmus alkalmazható olyan nemlineáris programozási feladatok megoldására is, amelyek csak egy, vagy néhány nemlineáris feltételt tartalmaznak, amennyiben a feltételi függvények és a célfüggvény eleget tesznek az alább felsorolt feltételeknek. Az algoritmus konvergenciája bizonyára gyengébb feltételek mellett is bizonyítható, mi az alábbiakat választottuk, mert egyrészt ezek teljesültek az általunk megoldott konkrét problémák esetében, másrészt pedig a konvergencia bizonyítása röviden tárgyalható. A teljesség kedvéért a feltételrendszer tartalmazza a $G(\mathbf{x})$ függvénynek a 2. részben bizonyított tulajdonságait is. Ezek után a feltételek:

a) Az $f(\mathbf{x})$ célfüggvény konkáv, differenciálható függvény, a gradiense *Lipschitz-folytonos*, azaz van olyan $L>0$ konstans, hogy minden $\mathbf{x}\in R^n$, $\mathbf{y}\in R^n$-re

$$\|\nabla f(\mathbf{x})-\nabla f(\mathbf{y})\| \leqq L\cdot\|\mathbf{x}-\mathbf{y}\|.$$

b) $G(\mathbf{x})$ logaritmikusan konkáv, differenciálható függvény, $\nabla G(\mathbf{x})$ *Lipschitz-folytonos* és korlátos R^n-en.

c) Van olyan megengedett $\mathbf{y}$ vektor, melyre $G(\mathbf{y})>p$ teljesül.

d) A (2.1) feladat megengedett megoldásainak halmaza korlátos.

e) Minden $\mathbf{x}$ megengedett ponthoz létezik az $\mathbf{A}$ mátrixnak olyan $\mathbf{B}$ bázisa, hogy ha $I_{\mathbf{B}}$ jelöli a bázisindexek halmazát, akkor $x_i>0$ minden $i\in I_{\mathbf{B}}$-re teljesül. Másszóval kizárjuk a degenerációt. Tételezzük fel továbbá, hogy $r(\mathbf{A})=m$, azaz az $\mathbf{A}$ mátrix teljes rangú.

Legyen $\mathbf{x}^{(1)}$ a (2.1) feladat egy megengedett megoldása, és tételezzük fel, hogy az $\varepsilon_1>0$ számot és az I_1 indexhalmazt úgy választottuk, hogy egyrészt az $\mathbf{A}$ mátrix I_1-be tartozó indexű oszlopai bázist alkossanak, másrészt $x_i^{(1)}>\varepsilon_1$ teljesüljön minden $i\in I_1$-re. Ez az e) feltétel miatt lehetséges. Legyenek továbbá $\vartheta>0$, $R>0$ az eljárás során rögzített számok.

Algoritmusunk iteratív, az egyes iterációk során új $\mathbf{x}^{(k)}$ megengedett megoldást, ε_k számot és I_k indexhalmazt határozunk meg, $k=1, 2, \ldots$. Az eljárás során az aktuális I_k indexhalmaznak megfelelő bázis inverzét tároljuk, és a báziscseréknek megfelelően transzformáljuk.

Tételezzük fel, hogy a $k=1$ esetnél leírt tulajdonságú $\mathbf{x}^{(k)}$, ε_k, I_k-t már kiszámoltuk, $k\geqq 1$. Az egyszerűbb jelölésmód kedvéért tegyük fel, hogy az $\mathbf{A}$ első m oszlopa alkotja a bázist, azaz $I_k=\{1, \ldots, m\}$, a bázisnak megfelelő mátrixot jelöljük $\mathbf{B}$-vel. Particionáljuk az $\mathbf{A}$ mátrixot és a megengedett megoldást bázis illetve bázison kívüli részre: $\mathbf{A}=(\mathbf{B}, \mathbf{C})$, $\mathbf{x}^{(k)}=(\mathbf{y}^{(k)}, \mathbf{z}^{(k)})$, hasonlóképp az irányvektort:

$\mathbf{w}^{(k)}=(\mathbf{u}^{(k)}, \mathbf{v}^{(k)})$, ahol $\mathbf{y}^{(k)}\in R^m$, $\mathbf{u}^{(k)}\in R^m$, $\mathbf{B}$ $m\times m$-es mátrix. Ekkor természetesen fennáll: $y_i^{(k)}>\varepsilon_k$, $i=1, \ldots, m$-re.

Ezek után $\mathbf{x}^{(k+1)}$, ε_{k+1}, I_{k+1}-et a következő lépéssorozattal határozzuk meg:

1. *Lépés.* Kiszámoljuk $f(\mathbf{x})$-nek és $G(\mathbf{x})$-nek a feltételekben szereplő lineáris egyenletrendszerre vonatkozó redukált gradiensét az $\mathbf{x}^{(k)}$ pontban:

$$\begin{aligned} \mathbf{r}^T &= \nabla_{\mathbf{z}}^T f(\mathbf{x}^{(k)}) - \nabla_{\mathbf{y}}^T f(\mathbf{x}^{(k)})\mathbf{B}^{-1}\mathbf{C} \\ \mathbf{s}^T &= \nabla_{\mathbf{z}}^T G(\mathbf{x}^{(k)}) - \nabla_{\mathbf{y}}^T G(\mathbf{x}^{(k)})\mathbf{B}^{-1}\mathbf{C}. \end{aligned} \tag{5.1}$$

2. *Lépés.* Megoldjuk a következő iránykereső feladatot:

$$\begin{aligned} &\max \zeta \\ &\mathbf{r}^T\mathbf{v} \geqq \zeta, \\ &\mathbf{s}^T\mathbf{v} \geqq \vartheta\zeta, \quad \text{ha} \quad G(\mathbf{x}^{(k)}) \leqq p+\varepsilon_k, \\ &v_i \geqq 0, \quad \text{ha} \quad z_i^{(k)} \leqq \varepsilon_k, \quad k=1, \ldots, n-m, \\ &\|\mathbf{v}\| \leqq 1. \end{aligned} \tag{5.2}$$

Jelöljük az (5.2) feladat optimális megoldását $(\hat{\mathbf{v}}, \hat{\zeta})$-val.

3. *Lépés.* Ha $\hat{\zeta}>\varepsilon_k$, akkor az eljárás a 4. lépéssel folytatódik.

Egyébként 2 esetet különböztetünk meg:

1. *eset.* $\hat{\zeta}\leqq\varepsilon_k$ de $\hat{\zeta}\neq 0$. Ekkor legyen $\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}$, $\varepsilon_{k+1}=\frac{1}{2}\varepsilon_k$, $I_{k+1}=I_k$ az iteráció befejeződött, $k:=k+1$, a következő iteráció a 2. lépésnél kezdődik.

2. *eset.* $\hat{\zeta}=0$. Megoldjuk a következő iránykereső feladatot:

$$\begin{aligned} &\max \zeta \\ &\mathbf{r}^T\mathbf{v} \geqq \zeta, \\ &\mathbf{s}^T\mathbf{v} \geqq \vartheta\zeta, \quad \text{ha} \quad G(\mathbf{x}^{(k)}) = p, \\ &v_i \geqq 0, \quad \text{ha} \quad z_i^{(k)} = 0, \quad i=1, \ldots, n-m, \\ &\|\mathbf{v}\| \leqq 1. \end{aligned} \tag{5.3}$$

Jelöljük az (5.3) feladat optimális megoldását $(\mathbf{v}_0, \zeta_0)$-lal. Újra két esetet különböztetünk meg. Ha $\zeta_0=0$, akkor az eljárás véget ért, $\mathbf{x}^{(k)}$ a (2.1) feladat optimális megoldása. Ha pedig $\zeta_0>0$, akkor az iteráció befejeződött, $\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}$, $\varepsilon_{k+1}=\frac{1}{2}\varepsilon_k$, $I_{k+1}=I_k$, $k:=k+1$, a következő iteráció a 2. lépésnél kezdődik.

4. *Lépés.* Legyen $\mathbf{v}^{(k)}=\hat{\mathbf{v}}$, $\mathbf{u}^{(k)}=-\mathbf{B}^{-1}\mathbf{C}\mathbf{v}^{(k)}$ és az irány: $\mathbf{w}^{(k)}=(\mathbf{u}^{(k)}, \mathbf{v}^{(k)})$

5. *Lépés.* Meghatározunk egy induló lépéshosszat:

$$\alpha_k = \begin{cases} \min\limits_{w_i^{(k)}<0} \left(-\dfrac{x_i^{(k)}}{w_i^{(k)}}\right), & \text{ha van olyan } i,\ 1\leqq i\leqq n, \text{ melyre } w_i^{(k)}<0, \\ R, & \text{különben.} \end{cases} \tag{5.4}$$

6. *Lépés.* Kiszámítjuk a λ_k lépéshosszot. Legyen $\lambda_k = \frac{1}{2^{l_0}} \alpha_k$, ahol $l = l_0$ az első egész szám, melyre a következő egyenlőtlenségek teljesülnek:

$$f\left(\mathbf{x}^{(k)} + \frac{\alpha_k}{2^l}\mathbf{w}^{(k)}\right) \geqq f(\mathbf{x}^{(k)}) + \frac{\alpha_k}{2^{l+1}}\zeta,$$

(5.5)

$$G\left(\mathbf{x}^{(k)} + \frac{\alpha_k}{2^l}\mathbf{w}^{(k)}\right) \geqq p,$$

$l \geqq 0$, ha (5.4)-ben α_k-t az első sor szerint számoltuk.

Más szóval vagy (5.4)-ben α_k-t az első sor szerint számoltuk és $l = l_0 = 0$ kielégíti (5.5)-öt, vagy pedig $l = l_0$ kielégíti (5.5)-öt, de $l = l_0 - 1$ nem.

7. *Lépés:*

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{w}^{(k)},$$

8. *Lépés:* Ha $x_i^{(k+1)} > \varepsilon_k$ minden $i \in I_k$-ra, akkor az iteráció befejeződött, $\varepsilon_{k+1} = \varepsilon_k$, $I_{k+1} = I_k$, $k := k+1$ a következő iteráció az 1. lépésnél kezdődik.

Egyébként az adott bázisból kiindulva olyan bázist keresünk, hogy a hozzá tartozó I_{k+1} indexhalmazra teljesüljön $x_i^{(k+1)} > \varepsilon_{k+1}$, ha $i \in I_{k+1}$, ahol $\varepsilon_{k+1} = \frac{1}{2^q}\varepsilon_k$ és és itt vagy $q = 0$, vagy q az első pozitív egész szám, melyre fenti tulajdonságú bázis található. Az e) feltétel miatt ilyen bázis található, ha ε_{k+1} elég kicsi. Ezek után az I_{k+1}-nek megfelelően transzformáljuk $\mathbf{B}^{-1}$-et, az iteráció befejeződött, $k := k+1$, az új iteráció az 1. lépéssel folytatódik.

A továbbiakban röviden diszkutáljuk az algoritmust.

5.1. LEMMA: Ha a k-adik iteráció során az iteráció 3. lépésében $\zeta_0 = 0$, akkor $\mathbf{x}^{(k)}$ a (2.1) feladat optimális megoldása.

Bizonyítás: Ha $\zeta_0 = 0$, akkor a következő feladat optimális célfüggvényértéke is 0:

$$\begin{aligned} &\max \zeta \\ &\nabla_{\mathbf{x}}^T f(\mathbf{x}^{(k)})\mathbf{w} \geqq \zeta, \\ &\nabla_{\mathbf{x}}^T G(\mathbf{x}^{(k)})\mathbf{w} \geqq \vartheta\zeta, \quad \text{ha} \quad G(\mathbf{x}^{(k)}) = p, \\ &\mathbf{A}\mathbf{w} = \mathbf{0}, \\ &v_i \geqq 0, \quad \text{ha} \quad z_i^{(k)} = 0, \quad i = 1, \ldots, n-m. \end{aligned} \tag{5.6}$$

Ezek után a [11] dolgozat megfelelő gondolatmenete szerint járunk el: A *Farkas-tételt* alkalmazzuk, és azt a tényt, hogy a c) feltétel az $f(\mathbf{x})$ célfüggvénynek megfelelő sorhoz pozitív szorzót biztosít, így (5.6)-ból a *Kuhn—Tucker-feltételek* megkaphatók. A 2.2. lemma szerint ezek az optimalitás elegendő feltételei.

Megmutatjuk, hogy az algoritmus monoton növő célfüggvényérték-sorozatot generál:

5.2. LEMMA. Az algoritmus vagy véges sok lépésben megtalálja (2.1) optimális megoldását, vagy megengedett megoldások olyan végtelen sorozatát generálja,

melyeken a célfüggvényértékek sorozata monoton nő, továbbá van olyan részsorozat, melyen a monotonitás szigorú.

Bizonyítás: A monotonitás az algoritmus konstrukciójából világos. A szigorú monotonitásra vonatkozó állítás abból következik, hogy az algoritmus nyilvánvaló módon egymás után csak véges sokszor szakadhat meg a 3. lépésnél, azaz van olyan részsorozat, melyben eljut a 4. lépésig, és itt a szigorú növekedés biztosított.

Mielőtt továbblépnénk, megjegyezzük, hogy a generált $\mathbf{w}^{(k)}$ irányvektorsorozat korlátos. Valóban, mivel az $\mathbf{A}$ mátrixnak csak véges sok különböző bázisa létezik, ezért van olyan $Q>0$, melyre:

$$\|\mathbf{w}^{(k)}\| = \|(-\mathbf{B}^{-1}\mathbf{C}\mathbf{v}^{(k)}, \mathbf{v}^{(k)})\| \leqq Q\|\mathbf{v}^{(k)}\| = Q.$$

Az algoritmus 6. lépésénél meg kell mutatnunk, hogy λ_k mindig választható az ott megkövetelt módon. Ez a következő állítás egyszerű következménye:

5.3. LEMMA: Ha $\lambda \leqq \frac{1}{2}\frac{\hat{\zeta}}{Q^2 L}$, akkor $f(\mathbf{x}^{(k)}+\lambda\mathbf{w}^{(k)}) \geqq f(\mathbf{x}^{(k)})+\frac{1}{2}\lambda\hat{\zeta}$.

Bizonyítás: A *Lagrange-féle középértéktételt* alkalmazzuk, felhasználva $\nabla f(\mathbf{x})$ *Lipschitz-folytonosságát* és $\hat{\zeta}$ definícióját:

$$f(\mathbf{x}^{(k)}+\lambda\mathbf{w}^{(k)}) = f(\mathbf{x}^{(k)})+\lambda\nabla_{\mathbf{x}}^T f(\mathbf{x}^{(k)})\mathbf{w}^{(k)}+\lambda[\nabla_{\mathbf{x}}^T f(\mathbf{x}^{(k)}+\Theta\lambda\mathbf{w}^{(k)})-\nabla_{\mathbf{x}}^T f(\mathbf{x}^{(k)})]\mathbf{w}^{(k)} \geqq$$

$$\geqq f(\mathbf{x}^{(k)})+\lambda\hat{\zeta}-\lambda^2 Q^2 L \geqq f(\mathbf{x}^{(k)})+\frac{1}{2}\lambda\hat{\zeta},$$

ha itt $\lambda \leqq \frac{1}{2}\frac{\hat{\zeta}}{Q^2 L}$; a Θ-ra érvényes: $0<\Theta<1$.

Néhány megjegyzés az (5.2), (5.3) segédfeladatokról. Ha a $\|\mathbf{v}\| = \max_{1\leqq j\leqq n-m} |v_i|$ normát használjuk, akkor (5.2), (5.3), lineáris programozási feladatok, melyekben 2 feltételi sor szerepel, és így megoldásuk nem jelent problémát. A módszer kézenfekvő módon kiterjeszthető több nemlineáris feltétel kezelésére, ha l nemlineáris feltételünk van, az (5.2), illetve (5.3)-nak megfelelő segédfeladatok lineáris programozási problémák $l+1$ sorral. Ez azt jelenti, hogy a módszer kevés nemlineáris feltétel esetén látszik hatékonynak.

Ha a feladatban nem szerepel nemlineáris feltétel, vagy (2.1)-ben a nemlineáris feltétel sohasem válik ε_k-aktívvá, akkor az algoritmus a redukált gradiens módszerbe megy át, ha pedig a változókra vonatkozó egyedi alsó-felső korlátokon kívül (2.1)-ben nem szerepel lineáris feltétel, akkor az algoritmus ZOUTENDIJK P1 módszerével esik egybe.

Az algoritmus első fázisa, azaz induló megoldás keresése [12]-höz hasonlóan úgy történik, hogy $G(\mathbf{x})$-et maximalizáljuk a (2.1)-ben szereplő lineáris mellékfeltételekkel. Ehhez magát az algoritmust használjuk, ami a fenti megjegyzés szerint ebben az esetben a redukált gradiens módszert jelenti. Természetesen az iterációkat csak addig folytatjuk, amíg a $G(\mathbf{x})$ valószínűség meg nem haladja az előírt p szintet. A következő részben bebizonyítjuk az algoritmus konvergenciáját, ez a c) feltétellel együtt az első fázis végességét adja. A redukált gradiens módszert először KLEINMICHEL [6] és SADOWSKI [15] módosították úgy, hogy megengedett pontokkal dolgozzon, ők azonban másképp közelítik meg a problémát és az általuk adott algoritmus a fentitől különböző.

6. Az algoritmus konvergenciája

Az 5. részben megadott algoritmus megengedett irányokkal dolgozik, így a konvergencia bizonyítása a megengedett irányos algoritmusokra kidolgozott, ismert elvek alapján történhet: [13]. A fenti technikát alkalmazó algoritmusoknál a konvergenciabizonyítások kritikus lépése annak megmutatása, hogy $\varepsilon_k \to 0$, $(k \to \infty)$. Ez a *Lipschitz-folytonosság*ra vonatkozó feltevéseink miatt könnyen belátható:

6.1. TÉTEL: Ha az algoritmus végtelen $\mathbf{x}^{(k)}$ pontsorozatot generál, akkor $\varepsilon_k \to 0$, $(k \to \infty)$.

Bizonyítás: Az 5.3. lemma gondolatmenetét megismételve, ha $G(\mathbf{x}^{(k)}) \leqq p + \varepsilon_k$ és egy egyszerű, a *Lagrange-középértéktétel*re épülő becslést alkalmazva a $G(\mathbf{x}^{(k)}) > p + \varepsilon_k$ esetben, megmutatható, hogy $G(\mathbf{x}^{(k)} + \lambda \mathbf{w}^{(k)}) \geqq p$, amennyiben $\lambda \leqq \min \cdot \left\{ \frac{\vartheta \hat{\zeta}}{Q^2 L}, \frac{\varepsilon_k}{K \cdot Q} \right\}$, ahol K a $\|\nabla G(\mathbf{x})\|$ egy felső korlátja. Az 5.3. lemma és a fentiek alapján világos, hogy ha

$$\lambda \leqq \min \left\{ \frac{1}{2} \frac{\hat{\zeta}}{Q^2 L}, \frac{\vartheta \hat{\zeta}}{Q^2 L}, \frac{\varepsilon_k}{KQ}, \frac{\varepsilon_k}{Q} \right\},$$

akkor $\mathbf{x}^{(k)} + \lambda \mathbf{w}^{(k)}$ megengedett megoldása (2.1)-nek és teljesül:

$$f(\mathbf{x}^{(k)} + \lambda \mathbf{w}^{(k)}) \geqq f(\mathbf{x}^{(k)}) + \frac{1}{2} \lambda \hat{\zeta}. \tag{6.1}$$

Az 5.2. lemma szerint van olyan részsorozat, ahol $\hat{\zeta}_k > \varepsilon_k$, itt $\hat{\zeta}_k$ jelöli $\hat{\zeta}$-ot a k-adik iterációban. Legyen a részsorozat indexeinek halmaza J. Ha $k \in J$, akkor $\mathbf{x}^{(k)} + \lambda \mathbf{w}^{(k)}$ megengedett megoldása (2.1)-nek és (6.1) is teljesül, amennyiben $\lambda \leqq \varepsilon_k \cdot \varepsilon_0$, ahol $\varepsilon_0 = \min \left\{ \frac{1}{2Q^2 L}, \frac{\vartheta}{Q^2 L}, \frac{1}{KQ}, \frac{1}{Q} \right\} > 0$. Az algoritmus 6. lépésénél l_0 az első olyan egész szám volt, melyre (5.5) teljesült, ezért szükségképpen $\lambda_k > \frac{1}{2} \varepsilon_k \varepsilon_0$, ha $k \in J$. Így 6.1-ből

$$f(\mathbf{x}^{(k+1)}) \geqq f(\mathbf{x}^{(k)}) + \frac{1}{4} \varepsilon_0 \cdot \varepsilon_k^2, \quad k \in J \tag{6.2}$$

adódik.

Mivel $f(\mathbf{x}^{(k)})$ monoton növő, korlátos sorozat, ezért $\varepsilon_k \to 0$, $k \in J$, $k \to \infty$. Felhasználva, hogy az ε_k-k sorozata monoton fogy, a tétel állítását bebizonyítottuk.

A 6.1. tétel felhasználásával a konvergencia bizonyítása:

6.2. TÉTEL. Az 5. rész algoritmusa vagy véges sok lépésben megtalálja (2.1) optimális megoldását, vagy a generált végtelen $\mathbf{x}^{(k)}$, $k = 1, 2, \ldots$ sorozat minden torlódási pontja a (2.1) feladat optimális megoldása.

Bizonyítás: Mivel $f(\mathbf{x})$ konkáv és a generált pontokra $f(\mathbf{x}^{(k)})$ monoton sorozat, ezért elegendő megmutatni, hogy az $\mathbf{x}^{(k)}$, $k = 1, 2, \ldots$, sorozatnak van olyan konvergens részsorozata, melynek határpontja a (2.1) feladat optimális megoldása. Mielőtt egy ilyen részsorozat kiválasztásának módját megadnánk, megjegyezzük, hogy az e) feltétel következményeképpen ε_k-t az iteráció 8. lépésében csak véges sok-

szor felezzük. Ezt a megjegyzést és a 6.1. tételt felhasználva világos, hogy van olyan részsorozat, melyre $\hat{\zeta}_k \leqq \varepsilon_k$ teljesül. A részsorozat indexei halmazát jelöljük J-vel. A d) feltevésből következően feltehetjük, hogy részsorozatunk konvergens, azaz van olyan $\mathbf{x}^*$ pont, hogy $\mathbf{x}^{(k)} \to \mathbf{x}^*$, $k \in J$, $k \to \infty$. Az $\mathbf{x}^*$ pont nyilvánvalóan megengedett megoldása (2.1)-nek. Tekintsük $\mathbf{x}^*$-ban a következő iránykereső feladatot:

$$
\begin{aligned}
&\max \zeta \\
&\nabla_{\mathbf{x}}^T f(\mathbf{x}^*)\mathbf{w} \geqq \zeta, \\
&\nabla_{\mathbf{x}}^T G(\mathbf{x}^*)\mathbf{w} \geqq \vartheta\zeta, \quad \text{ha} \quad G(\mathbf{x}^*) = p, \\
&\mathbf{A}\mathbf{w} = \mathbf{0} \\
&w_i \geqq 0, \quad \text{ha} \quad x_i^* = 0, \quad i = 1, \ldots, n, \\
&\|\mathbf{w}\| \leqq 1.
\end{aligned} \tag{6.3}
$$

Jelöljük a (6.3) feladat optimális megoldását $(\mathbf{w}^*, \zeta^*)$-gal. Ha itt $\zeta^* = 0$, akkor az 5.1. lemma bizonyítási technikájával könnyen adódik, hogy $\mathbf{x}^*$ optimális megoldása (2.1)-nek. Tételezzük fel, hogy $\zeta^* > 0$, ellentmondásra fogunk jutni. Ha $G(\mathbf{x}^*) > p$, akkor elég nagy k-ra $G(\mathbf{x}^{(k)}) > p + \varepsilon_k$, $k \in J$. Hasonlóképp ha valamely i-re, $1 \leqq i \leqq n$, $x_i^* > 0$ teljesül, akkor elég nagy k-ra $x_i^{(k)} > \varepsilon_k$, $k \in J$.

Másrészt $\nabla f(\mathbf{x})$ és $\nabla G(\mathbf{x})$ folytonossága miatt elég nagy k-ra:

$$
\nabla_{\mathbf{x}}^T f(\mathbf{x}^{(k)})\mathbf{w}^* \geqq \frac{1}{2}\zeta^*,
$$

$$
\nabla_{\mathbf{x}}^T G(\mathbf{x}^{(k)})\mathbf{w}^* \geqq \frac{1}{2}\vartheta\zeta^*, \quad \text{ha} \quad G(\mathbf{x}^*) = p.
$$

Tekintsük továbbá $\mathbf{A}$ összes lehetséges bázisát, jelölje t a $\mathbf{w}^*$ megfelelő bázison kívüli részei normáinak maximumát. Az eddigiekből következik, hogy elég nagy k-ra, $k \in J$, $\left(\frac{1}{t}\mathbf{w}^*, \frac{1}{2t}\zeta^*\right)$ megfelelő bázison kívüli része megengedett megoldása az (5.2) iránykereső feladatnak. Ez azt jelenti, hogy van olyan k_0 index, hogy $k > k_0$, $k \in J$-re $\hat{\zeta}_k \geqq \frac{1}{2t}\zeta^*$, ami ellentmond annak, hogy $k \in J$-re $\hat{\zeta}_k \leqq \varepsilon_k$ és így a 6.1. tétel miatt $\hat{\zeta}_k \to 0$, $k \in J$, $k \to \infty$.

Ezzel a konvergencia bizonyítását befejeztük.

7. Numerikus tapasztalatok az új algoritmussal kapcsolatban

Az algoritmus tesztelésére program íródott az MTA CDC 3300-as gépére, FORTRAN nyelven. A program az algoritmus két fázisát egybeépítve tartalmazza, inputként egy, a lineáris feltételeket kielégítő vektor és a hozzá tartozó, megfelelő bázisindexek adandók meg. Ilyen induló megoldást a CDC *lineáris programozási package*-ét, a REX-et használva könnyen sikerült találni.

Az algoritmust először kisméretű feladatokon teszteltük. Az új módszer valamennyi feladatra jobbnak bizonyult a redukált gradiens módszer specializált vál-

tozatánál. A $G(\mathbf{x})$ értékét a feladatok egy részénél DEÁK ISTVÁN szubrutinjával számoltuk, más részénél egy egyszerű numerikus integráló rutinnal, ami $r=3$-ig kielégítően dolgozik. Az alábbiakban ismertetett futási eredmények egy kétváltozós feladatra vonatkoznak, a sztochasztikus feltételben is két sor van, $r=2$. A feladat a következő, DEÁK [2];

$$\min (x_1+x_2)$$

$$P\begin{Bmatrix}2x_1+x_2-6 \geqq \beta_1\\ x_1+8x_2-8 \geqq \beta_2\end{Bmatrix} \geqq 0{,}8 \tag{7.1}$$

$$x_1+4x_2 \geqq 4$$

$$3x_1+x_2 \geqq 3$$

$$x_1, x_2 \geqq 0,$$

ahol β_1, β_2 együttes eloszlása normális, várható értékük 0, szórásuk 1 és a korrelációs együttható R.

A feladat megoldása során kapott számolási eredmények jól tükrözték SLEPIAN következő tételét: A kétdimenziós normális eloszlásfüggvény, $\Phi(x_1, x_2; R)$ fix x_1, x_2-re az R korrelációs együttható monoton növő függvénye. Ez a (7.1) feladattal kapcsolatban azt jelenti, hogy növekvő R esetén az optimális célfüggvényértékek csökkennek. Ez jól látható a megfelelő futási eredményeket tartalmazó 1. táblázatban; ahol az első oszlop a korrelációs együtthatót, a 2. és 3. oszlop a (7.1) feladatra a program által talált optimális megoldás komponenseit, a 4. oszlop az ehhez tartozó célfüggvényértéket tartalmazza.

1. TÁBLÁZAT

R	x_1	x_2	$f_{opt.}$
−0,9	1,99743	0,97838	2,97590
−0,6	1,99524	0,98034	2,97558
−0,4	1,99527	0,97947	2,97473
−0,2	1,99091	0,97648	2,96739
0,0	1,99003	0,97314	2,96317
0,2	1,98782	0,95787	2,95571
0,4	1,99699	0,95908	2,94607
0,6	1,99323	0,93855	2,93178
0,9	1,99193	0,90068	2,89261

Miután a módszert kisméretű feladatokon alaposan teszteltük, megoldottuk a [12] dolgozatban ismertetett, a negyedik ötéves terv villamosenergiaipari ágazatára vonatkozó modell egy változatát is. Miután SZÁNTAI vizsgálataiból kiderült [16], hogy a sztochasztikus feltételben szereplő 4 sor közül a 2. és a 4. játszik lényeges szerepet, a modell egy olyan változatát oldottuk meg, amelyben csak ez a két sor szerepel, a korrelációs együttható 0,1. A (2.1) feladat jelöléseivel $m=50$, $n=72$ volt, a feladatot lefuttattuk $p=0{,}90$ és $p=0{,}95$-ös szinten is.

Az optimális megoldás azon komponenseit tartalmazza a 2. táblázat, melyek a sztochasztikus feltételben szerepelnek.

2 TÁBLÁZAT

$p=$	0,90	0,95
x_1	6342,93	6406,05
x_2	12787,19	12800,79
x_6	4562,13	4562,13
x_7	40,52	40,50
x_{24}	18400,00	18400,00
x_{26}	24,14	24,13

A feladat futásának néhány további jellemzőjét tartalmazza a következő táblázat:

3. TÁBLÁZAT

		0,90	0,95
$f_{\text{induló}}$	0,49	3875,00	3875,00
$f_{\text{optimális}}$	0,99	4371,68	4370,73
sztochasztikus feltétel az opt.-ban	0,99	0,89	0,95
iterációk száma	19	18	16
sztochasztikus feltétel kiszámolásainak száma	37	68	61
báziscserék száma	18	18	16
számolási idő	51,24 sec	45,71 sec	40,82 sec
	első fázis	második fázis	

Megadjuk a célfüggvényértékek növekedését is a $p=0,90$ esetben.
Első fázis:

0,4999, 0,5060, 0,5127, 0,5137, 0,5234, 0,5338, 0,5338,
0,5395, 0,5405, 0,5420, 0,5498, 0,5692, 0,5721,
0,6049, 0,7250, 0,7721, 0,8036, 0,8793, 0,9964.

A kezdeti lassú növekedés attól van, hogy az indulópont kedvezőtlen a módszer szempontjából, több bázisváltozó túl közel van a korlátjához. Miután ezek a bázisból kikerülnek, az eljárás felgyorsul. Az első fázis végeredménye jó indulópont a második fázis részére:
Második fázis:

3875,00, 4255,32, 4295,03, 4312,07, 4336,19,
4366,85, 4366,86, 4367,09, 4368,23, 4370,30,
4371,08, 4371,42, 4371,53, 4371,66, 4371,68,
4371,68, 4371,68, 4371,68.

Anélkül, hogy ebből különösebb következtetéseket kívánnánk levonni, pusztán azért, hogy a közölt számolási időkkel kapcsolatban valamilyen támpontot nyújtsunk, megemlítjük, hogy a REX lineáris programozási csomag a fenti villamosenergiaipari modell determinisztikus változatát 51 sec alatt oldotta meg két fázisban.

IRODALOM

[1] ABADIE, J. and CARPENTIER, J., «Généralisation de la méthode du gradient réduit de Wolfe au cas des contraintes non-linéaires», *Proc. IFØRS Conf.* Ed. D. B. Hertz és J. Melese, (Wiley, New York, 1966) 1041—1053.
[2] DEÁK, I. „Egy sztochasztikus programozási modell számítógépes kiértékelése", *MTA Számítástechnikai Központ Közlemények* **9** (1972) 33—48.
[3] DEÁK, I., "Monte-Carlo evaluation of the multidimensional normal distribution function by the ellipsoid method", *Problems of Control and Information Theory*, megjelenés előtt.
[4] FAURE, P., HUARD, P., «Resolution des programmes mathematiques á fonction non linéaire par la méthode du gradient réduit», *Revue Française de Recherche Operationelle* **36** (1965) 167—206.
[5] GERENCSÉR, L., Nemlineáris programozási feladatok megoldása szekvenciális módszerekkel, Kandidátusi disszertáció, Budapest, 1977.
[6] KLEINMICHEL, H., "On methods of feasible directions", előadás a III. Mátrafüredi Matematikai Programozási Konferencián, 1975.
[7] MAYER, J., "Computational experiences with the reduced gradient method", in: *Progress in Op. Res.* Ed. A. Prékopa (North Holland, 1974) 613—624.
[8] PRÉKOPA, A., "On probabilistic constrained programming", in: *Proc. of the Princeton Symp. on Math. Progr.* (Princeton Univ. Press, Princeton, N. Y. 1970) 113—138.
[9] PRÉKOPA, A., "Logarithmic concave measures with application to stochastic programming", *Acta Sci. Math.* **32** (1971) 301—316.
[10] PRÉKOPA, A., "Contributions to the theory of stochastic programming", *Mathematical Programming* **4** (1973) 202—221.
[11] PRÉKOPA, A., „A megengedett irányok elnevezésű nemlineáris programozási módszer kiterjesztése kvázikonkáv feltételi függvények esetére", *MTA Számítástechnikai Központ Közleményei* **9** (1972) 3—16.
12] PRÉKOPA, A., DEÁK, I., GANCZER, S. és PATYI, K., „A STABIL sztochasztikus programozási modell és annak kísérleti alkalmazása a magyar villamosenergia-iparra", *Alkalmazott Matematikai Lapok* **1** (1975) 3—22.
[13] Пшеничный, Б. Н., Данилин, Ю. М., *Численные методы в экстремальных задачах* (Наука, 1976).
[14] RAPCSÁK, T., Egy tározási modell számítástechnikai megoldása, Egyetemi doktori disszertáció, Debrecen, 1974.
[15] SADOWSKI, H., «Untersuchungen zu den Verfahren der reduzierten Gradienten in der nichtlinearen Optimierung», Doktori disszertáció, TU Dresden, 1973.
[16] SZÁNTAI, T., „A Prékopa-féle STABIL sztochasztikus programozási modell numerikus megoldásáról", *Alkalmazott Matematikai Lapok* **2** (1976) 93—101
[17] WOLFE, P., "Methods of nonlinear programming", in: *Recent Advances in Math. Progr.*, Ed. R. L. Graves and P. Wolfe McGraw-Hill, New York (1963) 76—77.
[18] ZOUTENDIJK, G., *Methods of Feasible Directions* (Elsevier Publ. Co., Amsterdam, New York, 1960).

(Beérkezett: 1977. július 27.)

MAYER JÁNOS
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST I., ÚRI U. 49.

ON THE STABIL STOCHASTIC PROGRAMMING MODEL

J. MAYER

In this paper we present a nonlinear programming algorithm for the solution of the STABIL stochastic programming model of A. PRÉKOPA. The method is a combination of *Zoutendijk's P1 method* and *the reduced gradient method of Wolfe, Abadie and Carpentier*. The convergence of the algorithm is proved and some computational experiences reported.

Alkalmazott Matematikai Lapok 2 (1976) 189—201

EGY ÚJ MÓDSZER SORBAKAPCSOLT TÁROZÓRENDSZER TERVEZÉSÉRE SZTOCHASZTIKUS PROGRAMOZÁS FELHASZNÁLÁSÁVAL

PRÉKOPA ANDRÁS, RAPCSÁK TAMÁS ÉS ZSUFFA ISTVÁN
Budapest

Sorbakapcsolt tározórendszerek optimális tervezésére vonatkozólag egy új modellt és megoldási módszert adunk. A modell leírását — tömör formában — már a [8] dolgozat is tartalmazza. Ebben a dolgozatban sokkal részletesebb tárgyalást adunk, a feladatot megoldjuk a szekvenciális, feltétel nélküli minimalizálási módszerrel (SUMT) és egy numerikus példát is adunk. Ez egy nagyobb feladat része volt és a módszer első hazai alkalmazása során merült fel.

1. Bevezetés

Az ebben a dolgozatban tárgyalt tározórendszer modell a [8] dolgozatban szerepel először. Ebben a dolgozatban a modellt részletesebben kifejtjük, megadjuk a felmerülő nemlineáris programozási feladat megoldási módszerét és bemutatunk egy numerikus példát.

Tározórendszerek optimális méretezésére és irányítására vonatkozó sztochasztikus programozási modelleket más szerzők korábban már publikáltak. Jó áttekintést ad ezekről a [4] összefoglaló dolgozat. Ezekben a modellekben szereplő valószínűségi feltételek azonban igen egyszerűek, nem együttes, hanem csak egyedi események valószínűségeit tartalmazzák. Ez azt jelenti, hogy a valószínűségi változók közötti függőséget elhanyagolják. Az ilyenformán felmerülő matematikai programozási probléma a hagyományos típusok közé tartozik, matematikai szempontból nem különbözik lényegesen attól a problémától, amit determinisztikus modell esetén nyernénk.

Az első tározómodell MORAN [5] nevéhez fűződik. Bár ez az első megfogalmazásban csak egy tározóra vonatkozik a modell általánosítható több tározó esetére is. Lényeges eleme azonban MORAN modelljének az, hogy a tározóba az egymás utáni periódusokban érkező vízmennyiségek független és azonos eloszlású valószínűségi változók, mert ezáltal lesz *Markov-lánc* a tározóban levő vízmennyiségek idősora. Ez a feltétel már csak kis mértékben enyhíthető. MORAN modellje ezen túlmenően nem törekszik gazdasági optimalizálásra, hanem csupán a folyamat valószínűségeinek a meghatározására, melyből pl. egy megbízhatósági jellegű feltétel kapcsán — hogy ti. a vízigényt elég nagy valószínűséggel mindig ki tudjuk elégíteni — bizonyos tározóméret, vagy méretek meghatározhatók.

Ebben a dolgozatban a fenti megszorító feltételek nem szerepelnek. Megengedünk sztochasztikusan összefüggő valószínűségi változókból alkotott és időben inhomogén vízmennyiség-idősort. Nemcsak a tározókba befolyó, hanem a vízigényeket képviselő mennyiségek is lehetnek valószínűségi változók. Ez utóbbiakra vonat-

kozólag szintén nem élünk függetlenségi és homogénitási megszorításokkal, sőt megengedjük, hogy a hozzáfolyási és a felhasználási sztochasztikus folyamatok is összefüggőek legyenek.

A modellel kapcsolatos matematikai vizsgálatok PRÉKOPA [6, 7, 8, 9, 10], logkonkáv mértékekkel kapcsolatos eredményeire támaszkodnak.

A dolgozatban — amint a cím is tanúsítja — sorbakapcsolt tározókkal foglalkozunk. Hasonló jellegű modellek megfogalmazhatók bonyolultabban kapcsolódó tározórendszerekre vonatkozólag is. Ha azonban a tározók nem sorbakapcsolt rendszert alkotnak, akkor már egy irányítási elvet is meg kell adnunk, azt ti., hogy amennyiben egy lentebb fekvő tározóban kevés víz van a felmerülő igényekhez képest, akkor mely tározókból és milyen mértékben kell a hiányt pótolnunk, feltéve, hogy az lehetséges. Látni fogjuk, hogy a sorbakapcsolt tározók esetén az optimális irányítási politika igen egyszerű, ha a víz értéke mindegyik tározónál ugyanaz.

A tározórendszer méretezési elv, melyet a dolgozatban tárgyalunk, abban áll, hogy előírjuk, miszerint bizonyos, egymás utáni periódusokban minden vízigény teljesíthető legyen egy bizonyos, a gyakorlatban 1-hez közeli valószínűséggel és ezt egy minimális létesítési, továbbá hiányköltséggel bíró rendszer esetében érjük el. A felhasznált víz elhagyja a rendszert, ezért tározóink öntözési, ipari és kommunális célokra szolgálnak és kizárjuk pl. a villamosenergia termelés esetét. Megjegyezzük, hogy hazai vonatkozásban a villamosenergiának csak igen kis mennyisége származik vízierőműtől.

A 2. szakaszban pontosan megmondjuk feltételeinket és megfogalmazzuk a modellt. A 3. szakaszban a modellt képviselő feladat numerikus megoldásával foglalkozunk. Végül a 4. szakaszban egy numerikus példát mutatunk be.

2. A feladat megfogalmazása

A folyók és a tározási lehetőségek topológiáját az 1. ábrán illusztráljuk.

Az időt egymás utáni periódusokra osztjuk és véges sok ilyen egymás utáni periódust veszünk figyelembe. E periódusok jelenthetnek hónapokat, dekádokat, heteket, stb.

Feltesszük, hogy minden egyes periódusban a hozzáfolyás a periódus elején megtörténik a vízrendszer topológiájának megfelelően: ha egy tározó megtelik, a víz túlfolyik rajta a lentebb fekvő tározókba, de minden egyes tározóban a lehető legtöbb vizet megtartjuk — ebben az időben — és csak a fölösleges mennyiséget engedjük tovább.

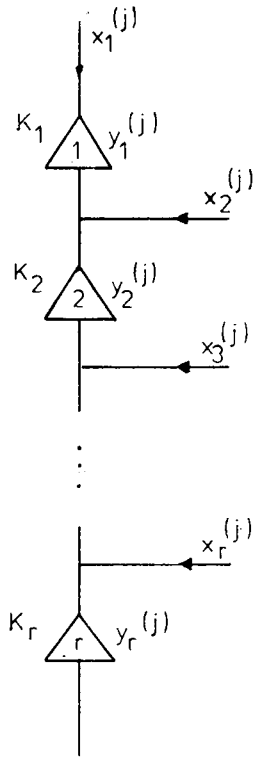
Feltesszük továbbá, hogy a vízigények mindig a periódusok végén jelentkeznek és a vízigények szétválaszthatók, az egyes tározókhoz hozzárendelhetők. Mindegyik vízigényt a megfelelő tározóból elégítjük ki, ha lehetséges. Ha nem lehetséges, akkor a következő irányítási elvnek megfelelően járunk el: előbb minden vízigényt oly mértékben kielégítünk, amilyen mértékben lehetséges; ezután a legalsó tározótól fölfelé haladva megállunk az első olyan tározónál, amely nem tudja kielégíteni a vele szemben támasztott vízigényt; innen kezdve és fölfelé haladva az egymás utáni tározók kielégítetlen vízigényeit összegyűjtjük, elmegyünk az első nem üres tározóig és megpróbáljuk annak vízfölöslegével ezt az összegezett vízigényt kielé-

gíteni; ha ez nem lehetséges, akkor hasonló értelemben tovább haladunk a felsőbb tározók felé.

Ha az egész rendszer képes a teljes vízigény kielégítésére, akkor a fent vázolt eljárás valahol megakad, kielégítjük a lenti tározók vízigényeit, majd az eljárást felfelé haladva megismételjük s. í. t. Az ismertetendő modellben a teljes vízigény kielégítésének a valószínűsége elég nagy lesz, tehát gyakorlatilag lehetséges lesz a teljes vízigény kielégítése.

Megjegyezzük, hogy amennyiben a víz értéke mindegyik tározó és mindegyik periódus esetében azonos, akkor a fenti tározórendszer irányítási elv nyilván optimális. Még azt is érdemes megemlíteni, hogy az ismertetett irányítási elv egyértelmű utasítást nyújt arra az esetre vonatkozólag is, amikor a teljes vízigény kielégítése nem lehetséges.

Feltesszük, hogy ha a tározórendszer valamely periódusban nem képes a teljes vízigény kielégítésére, akkor egy hiányköltség jelentkezik, mely az egész rendszerhez — és nem külön-külön az egyes tározókhoz — tartozik és arányos a hiány nagyságával. Az arányossági tényező függhet az időtől. Nemlineáris büntetőfüggvény alkalmazása is minden további nélkül lehetséges.



1. ábra
Sorbakapcsolt tározórendszer

Vezessük be a következő jelöléseket:

r a tározási lehetőségek száma;

K_i az i-edik tározó ismeretlen kapacitása;

V_i előírt szám, K_i felső korlátja;

n a figyelembe vett periódusok száma;

$z_i^{(0)}$ az i-edik tározó kezdő vízkészlete;

$z_i^{(j)}$ az i-edik tározó vízkészlete a j-edik periódus végén, feltéve, hogy az n periódus során nincs teljesítetlen vízigény; $z_i^{(j)}$ negatív értékeket is felvehet majd, a vonatkozó fizikai értelmezést később adjuk majd meg;

$x_i^{(j)}$ az i-edik tározóba a j-edik periódus elején közvetlenül befolyó vízmennyiség;

$y_i^{(j)}$ az i-edik tározóhoz hozzárendelt vízigény a j-edik periódusban;

$q^{(j)}$ egységnyi hiány költsége a j-edik periódusban;

$c_i(K_i)$ az i-edik tározó létesítési költsége a kapacitás függvényében;

ahol $i=1, \ldots, r$; $j=1, \ldots, n$.

Az $x_i^{(j)}$ és $y_i^{(j)}$ mennyiségeket valószínűségi változóknak tételezzük fel. A $z_i^{(j)}$ mennyiségek ezeknek és a $z_i^{(0)}$, K_i mennyiségeknek a függvényei, tehát ezek is valószínűségi változók. Egy pillanatra tételezzük fel azonban, hogy mennyiségeink determinisztikusak és nem sztohasztikusak, vagy ami ugyanaz, fixáljuk a valószínűségi változókat bizonyos lehetséges értékeiknél. Rekurzív relációkat fogunk felírni.

Először bevezetjük a $g_i^{(j)}$, $h_i^{(j)}$ mennyiségeket az alábbi egyenlőségekkel:

$$g_0^{(j)} = 0,$$

$$g_i^{(j)} = z_i^{(j-1)} + g_{i-1}^{(j)} + x_i^{(j)} - \min\,(z_i^{(j-1)} + g_{i-1}^{(j)} + x_i^{(j)},\ K_i), \tag{2.1}$$

$$h_i^{(j)} = \min\,(z_i^{(j-1)} + g_{i-1}^{(j)} + x_i^{(j)},\ K_i),$$

$$i = 1, \ldots, r;\ j = 1, \ldots, n.$$

$g_i^{(j)}$ az a vízmennyiség, amely túlfolyik az i-edik tározón a j-edik periódus elején és $h_i^{(j)}$ az i-edik tározóban maradó vízmennyiség ugyanennek a periódusnak az elején. Ezután értelmezzük a $d_i^{(j)}$ mennyiségeket a következő módon:

$$d_i^{(j)} = h_i^{(j)} - y_i^{(j)}, \quad i = 1, \ldots, r; \quad j = 1, \ldots, n. \tag{2.2}$$

Ezeket felhasználva, végül az alábbi relációkat írjuk fel:

$$z_1^{(j)} = \min\,(d_1^{(j)},\ d_1^{(j)} + d_2^{(j)},\ \ldots,\ d_1^{(j)} + \ldots + d_r^{(j)}),$$

$$z_i^{(j)} = \max\,[0,\ \min\,(d_i^{(j)},\ d_i^{(j)} + d_{i+1}^{(j)},\ \ldots,\ d_i^{(j)} + \ldots + d_r^{(j)})], \tag{2.3}$$

$$i = 2, \ldots, r; \quad j = 1, \ldots, n.$$

A $g_i^{(j)}$, $h_i^{(j)}$ mennyiségek fentebb megadott fizikai értelmezése csak akkor korrekt, ha fennáll az, hogy $z_1^{(j)} \geqq 0, j = 1, \ldots, n$. Ebben az esetben $z_i^{(j)}$ az i-edik tározó vízkészlete a j-edik periódus végén, minden $i = 1, \ldots, r$ és $j = 1, \ldots, n$ esetén.

Ha $z_1^{(j)} < 0$, akkor $-z_1^{(j)}$ a j-edik periódus végéig az egész rendszerben összegyűlt vízhiányt jelenti. Ha ez valamely j esetén bekövetkezik, akkor a rekurzív relációkban szereplő mennyiségek (legalábbis egy részének) korábban megadott fizikai értelmezése elvész, ám a rekurzív relációkat akkor is változatlan formában felhasználjuk, mennyiségeinket formálisan értelmezzük. Szükségünk van ugyanis ezekre többek között a $z_1^{(j)}$ mennyiségek értelmezése céljából. Az utóbbiak bekerülnek a célfüggvénybe.

Az n periódus során minden igényt akkor és csak akkor elégítünk ki, ha

$$z_1^{(j)} = \min\,(d_1^{(j)},\ d_1^{(j)} + d_2^{(j)},\ \ldots,\ d_1^{(j)} + \ldots + d_r^{(j)}) \geqq 0, \quad j = 1, \ldots, n. \tag{2.4}$$

Ez ekvivalens az alábbi egyenlőtlenségrendszerrel:

$$d_1^{(j)} \geqq 0, \quad d_1^{(j)} + d_2^{(j)} \geqq 0, \ \ldots, \ d_1^{(j)} + \ldots + d_r^{(j)} \geqq 0, \quad j = 1, \ldots, n. \tag{2.5}$$

A fenti relációkban csupán $z_1^{(0)}$, $x_i^{(j)}$, $y_i^{(j)}$, K_i, $i = 1, \ldots, r$; $j = 1, \ldots, n$ független változók (a determinisztikus értelemben), a többiek ezek segítségével kifejezhetők. E független változók száma $2r + 2rn$. A $d_i^{(j)}$, $z_i^{(j)}$, $g_i^{(j)}$, $h_i^{(j)}$ mennyiségek a fenti $2r + 2rn$ számú változó függvényei, de egyes független változók egyes függvénykapcsolatokban nem fordulnak elő. A $K_1, \ldots, K_r$ változókról feltesszük, hogy nemnegatívak, a többi, a (2.1)—(2.3) relációkban előforduló változókkal kapcsolatban azonban ilyen feltétellel nem élünk. Ezekre vonatkozólag azonban következnek bizonyos megszorítások a feladat feltételeiből. Most megfogalmazzuk a tározórendszer tervezésére vonatkozó feladatot. Ez a következő nemlineáris programozási

probléma:

$$\text{minimalizálandó } \left[\sum_{i=1}^{r} c_i(K_i) + \sum_{j=1}^{n} q^{(j)} E(\mu^{(j)})\right] \tag{2.6}$$

feltéve, hogy $P(d_1^{(j)} \geqq 0,\ d_1^{(j)} + d_2^{(j)} \geqq 0,\ \ldots,\ d_1^{(j)} + \ldots + d_r^{(j)} \geqq 0,\ j = 1, \ldots, n) \geqq q,$

$$0 \leqq K_i \leqq V_i, \quad i = 1, \ldots, r,$$

ahol q egy előírt valószínűség $(0 < q < 1)$, mely a gyakorlatban 1-hez közeli szám, $q^{(1)}, \ldots, q^{(n)}$ nemnegatív számok, melyek esetleg diszkontáló faktorokat tartalmaznak, ha viszonylag hosszú időről van szó, E a várható érték jele és

$$\mu^{(j)} = \begin{cases} -z_1^{(j)}, & \text{ha } z_1^{(j)} < 0, \\ 0, & \text{különben,} \end{cases} \quad j = 1, \ldots, n. \tag{2.7}$$

Feltesszük, hogy $c_i(K_i)$ a K_i változó folytonos függvénye a $[0, V_i]$ intervallumban, $i = 1, \ldots, r$.

3. A tározórendszer méretezési modell matematikai tulajdonságai

A (2.6) feladat olyan nemlineáris programozási feladat, melyben a célfüggvény és egy feltételi függvény nemlineáris. Különös figyelmet kell szentelnünk e feltételi függvénynek, melynek értékei többdimenziós térben elhelyezkedő halmazok valószínűségei. Ezek a valószínűségek a $K_1, \ldots, K_r$ ismeretlen kapacitásoktól, a feladat döntési változóitól függenek. Két fontos szempont is indokolja különös figyelmünket. Ezek a következők:

a) A (2.6) feladat numerikus megoldása akkor reményteljes, ha az egy konvex programozási feladat (konvex függvény minimalizálása konvex halmazon). Minthogy a (2.6) feladat célfüggvénye igen általános feltételek teljesülése esetén konvex, ezért a feladat konvex jellegét lényegében e feltételi függvény viselkedése szabja meg.

b) Bármilyen optimalizálási eljárást választunk is a (2.6) feladat megoldására, a megoldás minden lépésében legalábbis a feltételi függvények és a célfüggvény bizonyos értékére (esetleg még a gradiensek értékére is) szükség van.

A nemlineáris feltételi függvény értékei — mint említettük — valószínűségek többdimenziós térben. E függvény értékeinek meghatározása bonyolult feladat; mi *Monte Carlo-technikát* alkalmaztunk erre a célra.

Feladatunk matematikai tulajdonságait a következő két tételből vezetjük le.

3.1. TÉTEL. Legyenek $h_1(\mathbf{u}, \mathbf{v}), \ldots, h_r(\mathbf{u}, \mathbf{v})$ az $\mathbf{u}$, $\mathbf{v}$ vektorértékű változók összes komponensének konkáv függvényei az egész téren. Legyen ξ olyan valószínűségi vektorváltozó, melynek ugyanannyi komponense van, mint $\mathbf{v}$-nek. Tekintsük az $\mathbf{u}$ változó alábbi függvényét:

$$P(g_1(\mathbf{u}, \xi) \geqq 0, \ldots, g_r(\mathbf{u}, \xi) \geqq 0). \tag{3.1}$$

Ha ξ komponenseinek együttes valószínűségeloszlása folytonos és az együttes sűrűségfüggvény logkonkáv, akkor a (3.1) függvény az $\mathbf{u}$ változó logkonkáv függvénye az $\mathbf{u}$ vektorok terében.

A tétel bizonyítása lényegében a [6] dolgozatban megtalálható. Pontosan ebben a formában a [7] dolgozat említi. Hasonló tételek találhatók a [9] dolgozatban.

3.2. TÉTEL. A $d_1^{(j)}, d_1^{(j)}+d_2^{(j)}, \ldots, d_1^{(j)}+\ldots+d_r^{(j)}$, $j=1, \ldots, n$ függvények a $2r+2rn$ változó konkáv függvényei.

Bizonyítás. Előrebocsátjuk azt a könnyen igazolható állítást, hogy konkáv függvények minimuma is konkáv függvény.

Az alábbi egyenlőségek P-re vonatkozó indukcióval láthatók be:

$$h_1^{(j)}+\ldots+h_p^{(j)} = \min\left(z_1^{(j-1)}+\ldots+z_p^{(j-1)}+x_1^{(j)}+\ldots+x_p^{(j)}, K_p+h_1^{(j)}+\ldots+h_{p-1}^{(j)}\right), \tag{3.2}$$

$$p = 2, \ldots, r; \quad j = 1, \ldots, n.$$

Innen adódik, hogy

$$d_1^{(j)}+\ldots+d_p^{(j)} = \tag{3.3}$$

$$= \min\left(z_1^{(j-1)}+\ldots+z_p^{(j-1)}+x_1^{(j)}+\ldots+x_p^{(j)}-y_1^{(j)}-\ldots-y_p^{(j)}, K_p-y_p^{(j)}+d_1^{(j)}+\ldots+d_{p-1}^{(j)}\right),$$

$$p = 2, \ldots, r; \quad j = 1, \ldots, n,$$

továbbá, hogy

$$z_1^{(j)}+\ldots+z_p^{(j)} = \min\left(d_1^{(j)}+\ldots+d_p^{(j)}, \ldots, d_1^{(j)}+\ldots+d_r^{(j)}\right), \tag{3.4}$$

$$p = 2, \ldots, r; \quad j = 1, \ldots, n.$$

A (3.2) relációkban $h_i^{(j)}$, a (3.3) relációkban $d_i^{(j)}$ nem szerepel. Ezek azonban a (2.1), (2.2) egyszerű alakkal bírnak. A (2.1), (2.2) egyenlőségekből következik, hogy $d_i^{(j)}$ konkáv minden $j=1, \ldots, n$ esetén. Felhasználva ennek a $j=1$ esetre vonatkozó speciális esetét, továbbá a (3.3) egyenlőségeket, p-re vonatkozó indukcióval könnyen adódik, hogy a

$$d_1^{(1)}+\ldots+d_p^{(1)}, \quad p = 1, \ldots, r$$

függvények konkávak.

Most bebizonyítjuk a tétel állítását j-re vonatkozó indukcióval. Tegyük fel, hogy a

$$d_1^{(j-1)}+\ldots+d_p^{(j-1)}, \quad p = 1, \ldots, r$$

függvények konkávak. Ekkor a (3.4) egyenlőség szerint konkávak az alábbi függvények is:

$$z_1^{(j-1)}+\ldots+z_p^{(j-1)}, \quad p = 1, \ldots, r.$$

Felhasználva azt, hogy $d_i^{(j)}$ konkáv függvény, a (3.3) egyenlőségekből adódik, hogy a

$$d_1^{(j)}+\ldots+d_p^{(j)}, \quad p = 1, \ldots, r$$

függvények is konkávak. Eszerint az állítás érvényes j-re is. Ezzel a tételt bebizonyítottuk.

A 3.1. és 3.2. tételek közvetlen következménye az alábbi

3.3. TÉTEL. Ha az $x_1^{(j)}, \ldots, x_r^{(j)}, y_1^{(j)}, \ldots, y_r^{(j)}$ valószínűségi változók együttes eloszlása folytonos és az együttes sűrűségfüggvényük logkonkáv, akkor az alábbi függvény

$$P(d_1^{(j)} \geqq 0, d_1^{(j)}+d_2^{(j)} \geqq 0, \ldots, d_1^{(j)}+\ldots+d_r^{(j)} \geqq 0, j = 1, \ldots, n) \tag{3.5}$$

a $z_1^{(0)}, \ldots, z_r^{(0)}, K_1, \ldots, K_r$ változók logkonkáv függvénye.

Megjegyzés. A 3.3. tétel maga után vonja, hogy ha elvégezzük a $z_1^{(0)}=K_1, \ldots$ $\ldots, z_r^{(0)}=K_r$ helyettesítéseket, azaz teljesen feltöltött tározókkal indulunk, akkor a (3.5) függvény a $K_1, \ldots, K_r$ változók logkonkáv függvénye.

A [6] dolgozatban szerepel néhány példa logkonkáv többdimenziós sűrűségfüggvényekre vonatkozólag. Mi most a többdimenziós normális eloszlást fogjuk alkalmazni. A nem-elfajult esetben az eloszlás folytonos és sűrűségfüggvénye a következő

$$f(\mathbf{u}) = \frac{\sqrt{|\mathbf{C}|}}{(2\pi)^{k/2}} e^{-\frac{1}{2}(\mathbf{u}-\boldsymbol{\mu})' \mathbf{C}^{-1}(\mathbf{u}-\boldsymbol{\mu})}, \quad \mathbf{u} \in R^k, \tag{3.6}$$

ahol $\mathbf{C}$ a kovariancia mátrix és $\boldsymbol{\mu}$ a várható értékek vektora. $\mathbf{C}$-ről itt feltételezzük, hogy pozitív definit. Ebből következik, hogy $\mathbf{C}^{-1}$ is az. Másfelől ismeretes, hogy egy pozitív definit mátrixszal bíró kvadratikus alak (szigorúan) konvex függvény. Ebből tehát következik, hogy $f(\mathbf{u})$ az egész téren logkonkáv függvény.

A célfüggvénnyel kapcsolatban az alábbi tételt bizonyítjuk be.

3.4. TÉTEL. Akármilyen is az $x_i^{(j)}, y_i^{(j)}$, $i=1, \ldots, r$; $j=1, \ldots, n$ valószínűségi változók eloszlása, az alábbi függvény

$$\sum_{j=1}^{n} q^{(j)} E(\mu^{(j)})$$

a $z_i^{(0)}, K_i$, $i=1, \ldots, r$ változók konvex függvénye.

Bizonyítás. A 3.2. tétel és a (3.4) egyenlőtlenség szerint a

$$-z_1^{(j)}, \quad j = 1, \ldots, n$$

függvények konvexek. Ebből következik, hogy a (2.7) egyenlőséggel értelmezett $\mu^{(j)}$, $j=1, \ldots, n$ függvények szintén konvexek.

Általában, ha egy konvex függvény bizonyos változóit valószínűségi változókkal helyettesítjük, akkor az így kapott valószínűségi változó várható értéke a determinisztikusnak megmaradt változók konvex függvénye. Ezt a $\mu^{(j)}, j=1, \ldots, n$ függvényekre alkalmazva, tételünk bizonyítást nyert.

Feladatunk numerikus megoldására a szekvenciális, feltétel nélküli optimalizálási módszert (SUMT) alkalmazzuk, mégpedig a nemlineáris feltételi függvény tulajdonságának megfelelően logaritmikus büntető függvénnyel. A SUMT módszer részletes kifejtését illetően a [2] könyvre utalunk. Mi itt csak a (2.6) feladat megoldásával foglalkozunk.

Először a (2.6) feladat feltételeit oly módon alakítjuk át, hogy a jobb oldalakon zérók álljanak. Majd megalkotjuk a következő függvényt

$$\begin{aligned} &\sum_{i=1}^{r} c_i(K_i) + \sum_{j=1}^{n} q^{(j)} E(\mu^{(j)}) - \\ &-t\Big\{\ln [P(d_1^{(j)} \geqq 0, \ldots, d_1^{(j)} + \ldots + d_r^{(j)} \geqq 0, j = 1, \ldots, n) - q] + \\ &+ \sum_{i=1}^{r} \ln K_i + \sum_{i=1}^{r} \ln (V_i - K_i)\Big\}, \end{aligned} \tag{3.7}$$

ahol t rögzített pozitív szám. E függvénynek van egy feltétel nélküli minimuma, melyet a függvény felvesz egy t-től függő, mondjuk $K(t)$ helyen. Választunk egy $t_1 > t_2 > t_3 > \ldots$ pozitív számokból alkotott sorozatot, melyre fennáll, hogy $t_p \to 0$, ha $p \to \infty$ és mini-

malizáljuk a (3.7) függvényt mindegyik t_p esetén. Amint p növekszik, a (3.7) függvény és egyben az eredeti feladat célfüggvénye is egyre jobban megközelíti az optimum értékét. Állításunkat az alábbi tételben pontosabban is megfogalmazzuk.

3.5. TÉTEL. Ha a (3.5) függvény értéke a $K_i = V_i$, $i = 1, \ldots, r$ helyen nagyobb, mint q, akkor amint a $t_1, t_2, \ldots$ sorozat zéróhoz tart, a (3.7) függvény minimum-értékei tartanak a (2.6) feladat minimum-értékéhez.

Bizonyítás. A [2] könyv több általános tételt közöl a SUMT módszer konvergenciáját illetően. A mi (2.6) feladatunk a következő típusú

$$\begin{aligned} &\text{minimalizálandó } f(\mathbf{K}) \\ &\text{feltéve, hogy} \qquad h_i(\mathbf{K}) \geqq 0, \quad i = 1, \ldots, m, \end{aligned} \tag{3.8}$$

ahol $f, h_1, \ldots, h_m$ minden $\mathbf{K} \in R^r$ esetén értelmezett és folytonos függvények. A megengedett megoldások

$$F = \{\mathbf{K} | h_i(\mathbf{K}) \geqq 0,\ i = 1, \ldots, m\} \tag{3.9}$$

halmaza esetünkben korlátos, tehát a feladatnak van optimális megoldása. A logaritmikus büntető függvény az alábbi alakot ölti:

$$V_t(\mathbf{K}) = f(\mathbf{K}) - t \sum_{i=1}^{m} \ln h_i(\mathbf{K}),$$

ahol t pozitív állandó. Legyen $\mathbf{K}(t)$ olyan vektor, mely minimalizálja az előbbi függvényt rögzített t mellett.

A [2] könyvben a szerzők bebizonyítják, hogy ha t_p egy zéróhoz tartó, pozitív számokból alkotott, csökkenő sorozat, akkor a $V_{t_p}(\mathbf{K}(t_p))$ sorozat tart a (3.8) feladat optimum-értékéhez, feltéve, hogy a (3.9) halmaz minden relatív belső pontja eleme az alábbi halmaznak

$$M = \{\mathbf{K} | h_i(\mathbf{K}) > 0,\ i = 1, \ldots, m\} \tag{3.10}$$

melyről feltesszük, hogy nem üres. Csupán arra van tehát szükség, hogy ellenőrizzük az M halmaz e tulajdonságainak meglétét.

A 3.5. tételben feltettük, hogy a (3.5) függvény értéke nagyobb, mint q a $\mathbf{K} = \mathbf{V}$ helyen. Minthogy a függvény folytonos, van olyan $\mathbf{K}_1$, melynek esetében a (2.6) feladat feltételei mind határozott egyenlőtlenséggel teljesülnek. Eszerint a (3.10) halmaz a mi esetünkben nem üres.

Legyen $\mathbf{K}$ az F halmaz tetszőleges relatív belső pontja a (2.6) feladat esetében. Ekkor triviálisan fennállnak a $0 < K_i < V_i$, $i = 1, \ldots, r$ egyenlőtlenségek. Meg kell még mutatnunk, hogy a (3.5) függvény a $\mathbf{K}$ helyen q-nál nagyobb értéket vesz fel. Jelölje $P(\mathbf{K})$ a (3.5) függvényt. A $\mathbf{K} = \mathbf{K}_1$ esetben a fentiek szerint $P(\mathbf{K}) > q$. Ha $\mathbf{K} \neq \mathbf{K}_1$, akkor a $\mathbf{K}_1$-ből $\mathbf{K}$ irányába haladó félegyenesnek vesszük egy olyan $\mathbf{K}_2$ pontját, mely határpontja a (2.6) feladat megengedett megoldásai halmazának. Alkalmas $0 < \lambda < 1$ esetén fennáll a

$$\mathbf{K} = \lambda \mathbf{K}_1 + (1 - \lambda)\mathbf{K}_2$$

egyenlőség. Még azt is tudjuk, hogy $P(\mathbf{K}_1) > q$, $P(\mathbf{K}_2) \geqq q$, továbbá P logkonkáv függvény. Ebből azonnal adódik, hogy

$$P(\mathbf{K}) \geqq [P(\mathbf{K}_1)]^{\lambda} [P(\mathbf{K}_2)]^{1-\lambda} > q.$$

Ezzel a tételt bebizonyítottuk.

A 3.5. tétel gyakorlati jelentősége abban áll, hogy elég nagy p esetén a $\mathbf{K}(t_p)$ vektort a (2.6) feladat optimális megoldásának tekinthetjük.

Rögzített $t=t_p$ esetén el kell végeznünk a (3.7) függvény feltétel nélküli minimalizálását. A 3.3. tétel feltételeinek teljesülése esetén ez a függvény konvex, feltéve, hogy a $c_1(K_1), \ldots, c_r(K_r)$ függvények konvexek. Valóban, a $P(\mathbf{K})$ függvény logkonkávitásából következik a $P(K)-q$ függvény logkonkávitása (ezt igen egyszerűen be lehet látni az aritmetikai átlag-geometriai átlag egyenlőtlenség alapján) azon a halmazon, amelyen az utóbbi nemnegatív; figyelembe véve, hogy a 3.4. tétel szerint a büntető tagok összege $\mathbf{K}$-nak konvex függvénye, a (3.7) függvény valóban konvex minden rögzített $t>0$ esetén.

A (3.7) függvény konvexitása kellemes a függvény minimalizálása szempontjából. Ám a függvény más szempontból mégis bonyolult. Pl. nem remélhetjük, hogy könnyen meg tudjuk a gradiensét határozni. Ezért olyan minimalizálási módszert kell választani, mely nem használ gradienst. Ilyen pl. a *Hooke—Jeeves módszer* [3]. Ezt alkalmazzuk, miközben a függvényben szereplő valószínűségeket szimulációval határozzuk meg. A felhasznált szimulációs módszer leírása [11]-ben található. A módszerrel kapcsolatos újabb eredmények DEÁKtól származnak [1]. A modell természetesen más valószínűségeloszlás esetén is alkalmazható. Ajánlható pl. a [10]-ben leírt multigamma eloszlás.

4. Numerikus példa

Az alábbi feladat része volt a modell első gyakorlati alkalmazását jelentő tározórendszer méretezési feladatnak. Három periódusunk van, mindegyik egy hónap, mégpedig: június, július, augusztus. Két lehetséges tározási hely van. A hozzáfolyást és a vízigényt jelentő valószínűségi változók száma 12. Ezek a következők:

$$x_1^{\text{jún}}, y_1^{\text{jún}}, x_2^{\text{jún}}, y_2^{\text{jún}}, x_1^{\text{júl}}, y_1^{\text{júl}}, x_2^{\text{júl}}, y_2^{\text{júl}}, x_1^{\text{aug}}, y_1^{\text{aug}}, x_2^{\text{aug}}, y_2^{\text{aug}}.$$

Feltesszük, hogy $z_1^{(0)}=K_1$, $z_2^{(0)}=K_2$. A várható értékek, a szórások és a korreláció mátrix a következők:

	Várható értékek	Szórások
$x_1^{\text{jún}}$	464822	186984
$y_1^{\text{jún}}$	215760	327120
$x_2^{\text{jún}}$	929644	373960
$y_2^{\text{jún}}$	112033	275890
$x_1^{\text{júl}}$	320576	266040
$y_1^{\text{júl}}$	433608	243600
$x_2^{\text{júl}}$	641152	532080
$y_2^{\text{júl}}$	396225	205450
x_1^{aug}	266040	234040
y_1^{aug}	484416	214368
x_2^{aug}	532080	511060
y_2^{aug}	407965	180796

$$\mathbf{R} = \begin{pmatrix} 1{,}00 & 0{,}10 & 0{,}80 & 0{,}05 & 0{,}60 & 0{,}12 & 0{,}50 & 0{,}06 & 0{,}40 & 0{,}06 & 0{,}30 & 0{,}03 \\ & 1{,}00 & 0{,}05 & 0{,}80 & 0{,}12 & 0{,}25 & 0{,}10 & 0{,}23 & 0{,}08 & 0{,}02 & 0{,}05 & 0{,}00 \\ & & 1{,}00 & 0{,}13 & 0{,}55 & 0{,}15 & 0{,}68 & 0{,}13 & 0{,}50 & 0{,}00 & 0{,}52 & 0{,}00 \\ & & & 1{,}00 & 0{,}15 & 0{,}20 & 0{,}13 & 0{,}18 & 0{,}06 & 0{,}00 & 0{,}06 & 0{,}00 \\ & & & & 1{,}00 & 0{,}10 & 0{,}80 & 0{,}09 & 0{,}70 & 0{,}00 & 0{,}65 & 0{,}00 \\ & & & & & 1{,}00 & 0{,}09 & 0{,}70 & 0{,}15 & 0{,}20 & 0{,}13 & 0{,}02 \\ & & & & & & 1{,}00 & 0{,}10 & 0{,}65 & 0{,}00 & 0{,}70 & 0{,}00 \\ & & & & & & & 1{,}00 & 0{,}13 & 0{,}18 & 0{,}10 & 0{,}20 \\ & & & & & & & & 1{,}00 & 0{,}10 & 0{,}80 & 0{,}08 \\ & & & & & & & & & 1{,}00 & 0{,}10 & 0{,}85 \\ & & & & & & & & & & 1{,}00 & 0{,}10 \\ & & & & & & & & & & & 1{,}00 \end{pmatrix},$$

ahol a valószínűségi változók sorrendje ugyanaz, mint az előbb. A fentieken kívül megadjuk még V_1, V_2 értékét és a $c_1(K_1)$, $c_2(K_2)$ függvényeket:

$$V_1 = 1\,500\,000 \text{ m}^3$$

$$V_2 = 2\,500\,000 \text{ m}^3$$

$$c_1(K_1) = \begin{cases} K_1, & \text{ha} \quad 0 \leqq K_1 \leqq 500\,000, \\ 500\,000 + 0{,}4(K_1 - 500\,000), & \text{ha} \quad K_1 > 500\,000, \end{cases}$$

$$c_2(K_2) = \begin{cases} 0{,}45K_2, & \text{ha} \quad 0 \leqq K_2 \leqq 1\,000\,000, \\ 450\,000 + 0{,}6(K_2 - 1\,000\,000), & \text{ha} \quad 1\,000\,000 \leqq K_2 \leqq 1\,500\,000, \\ 750\,000 + 0{,}8(K_2 - 1\,500\,000), & \text{ha} \quad K_2 > 1\,500\,000, \end{cases}$$

ahol a költség egysége 100 Ft. A hiányköltség alkalmazásától eltekintünk. Végül a q valószínűség értéke legyen 0,8. Ezek után a tározórendszer méretezési feladat a következő:

$$\text{minimalizálandó } [c_1(K_1) + c_2(K_2)]$$

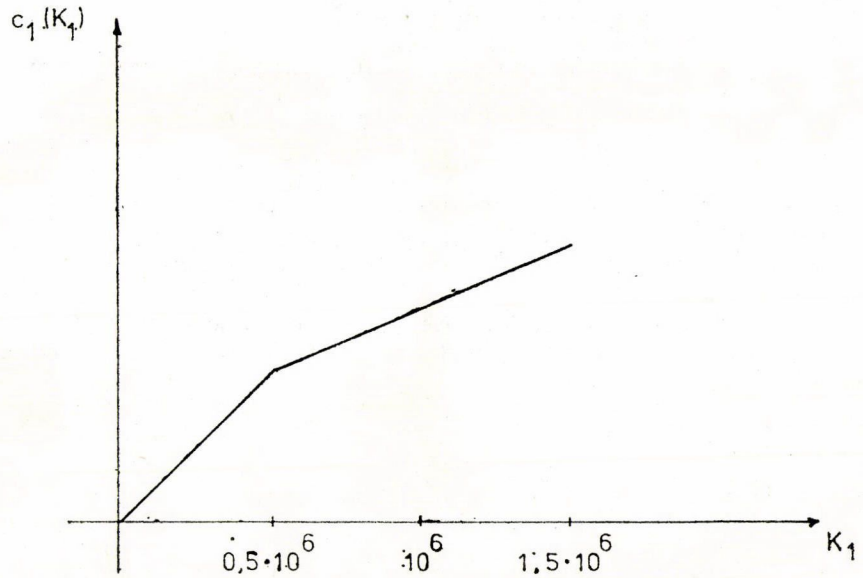
feltéve, hogy

$$P\begin{pmatrix} d_1^{(1)} \geqq 0, & d_1^{(1)} + d_2^{(1)} \geqq 0 \\ d_1^{(2)} \geqq 0, & d_1^{(2)} + d_2^{(2)} \geqq 0 \\ d_1^{(3)} \geqq 0, & d_1^{(3)} + d_2^{(3)} \geqq 0 \end{pmatrix} \geqq 0{,}8,$$

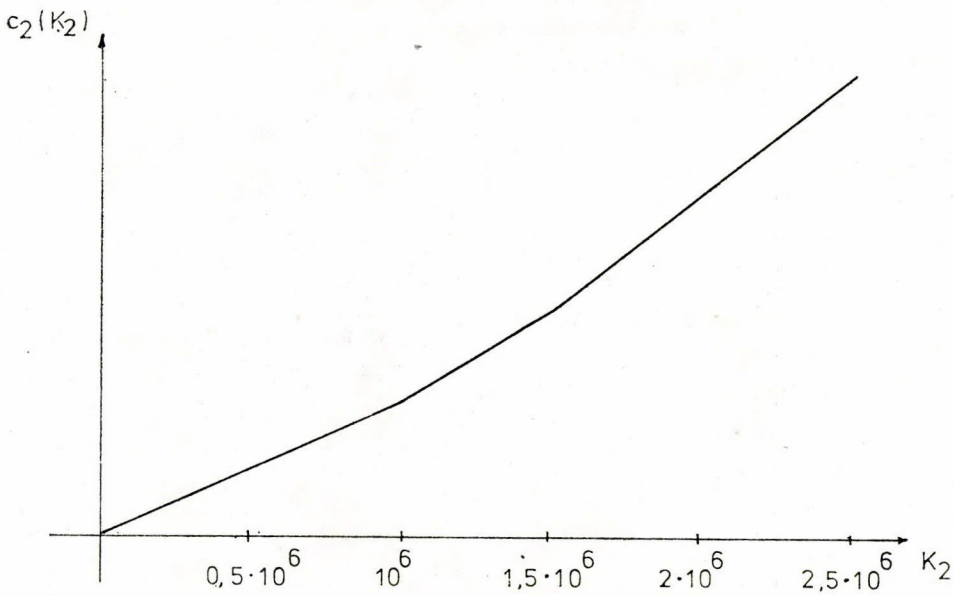
$$0 \leqq K_1 \leqq V_1,$$

$$0 \leqq K_2 \leqq V_2.$$

Vegyük észre, hogy $c_1(K_1)$ nem konvex függvény. Minthogy azonban a $0 \leqq K_1 \leqq \leqq 0{,}5 \cdot 10^6$ feltételnek eleget tevő K_1 értékek még a $K_2 = V_2$ esetben sem teljesítik a valószínűségi feltételt, feltehetjük, hogy $K_1 \geqq 0{,}5 \cdot 10^6$.



2. ábra
A $c_1(K_1)$ függvény ábrája. A függvény konkáv a [0, ∞) félegyenesen, de lineáris, tehát konvex a [0,5·10⁶, ∞) félegyenesen



3. ábra
A $c_2(K_2)$ függvény ábrája. A függvény konvex a [0, ∞) félegyenesen

A K_1-re ily módon meghatározott intervallumon $c_1(K_1)$ lineáris, tehát konvex függvény.

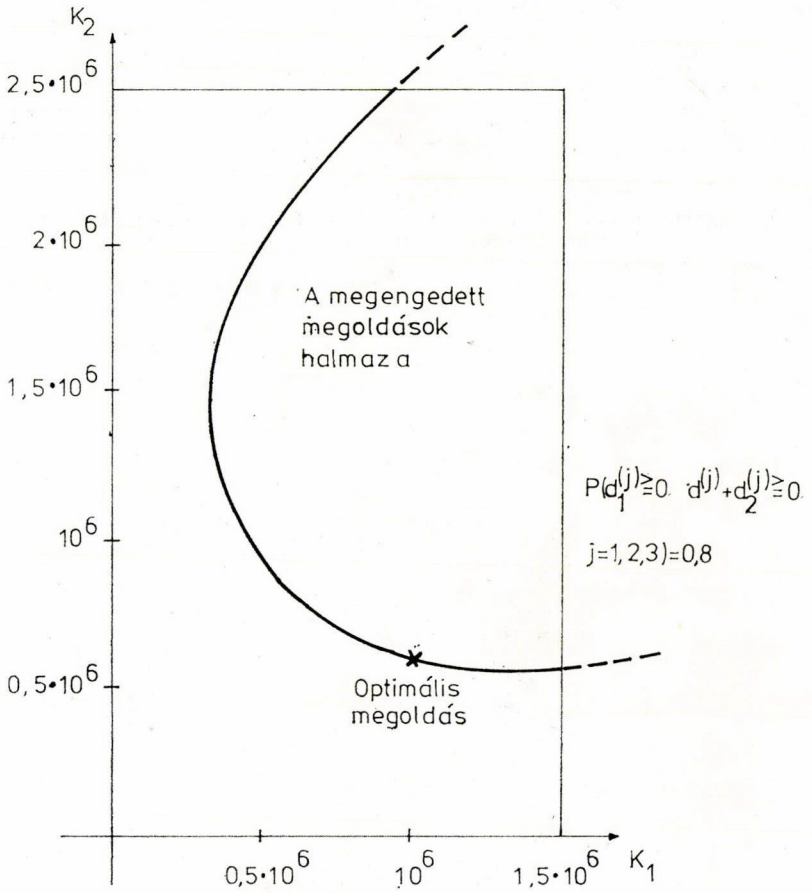
A feltétel nélküli minimalizálást az alábbi függvényre kell végrehajtanunk:

$$c_1(K_1)+c_2(K_2)-t\{\ln(P-0{,}8)+\ln K_1+\ln K_2+ \\ +\ln(1{,}5\cdot 10^6-K_1)+\ln(2{,}5\cdot 10^6-K_2)\}. \tag{4.2}$$

A feltétel nélküli minimalizálást háromszor hajtottuk végre a $t_1=1$, $t_2=1/5$, $t_3=1/25$ értékek esetében. Az induló megoldás a következő volt

$$K_1 = 1{,}4\cdot 10^6; \quad K_2 = 2{,}4\cdot 10^6.$$

A valószínűségi feltételben szereplő függvény értéke ebben az esetben 0,984. A második (harmadik) feltétel nélküli optimalizálás az első (második) feladat optimális megoldásából indult.



4. ábra
A megengedett megoldások halmazának szemléltetése

A *Hooke—Jeeves-módszer* 7, 2, 1 iterációt hajtott végre. A leállási szabályt az optimális megoldások komponensei eltérése segítségével fogalmaztuk meg. Optimális megoldásként a következőt kaptuk:

$$K_{1\,\mathrm{opt}} = 1{,}046\,289 \cdot 10^6\ \mathrm{m}^3,$$

$$K_{2\,\mathrm{opt}} = 0{,}611\,206 \cdot 10^6\ \mathrm{m}^3.$$

E kapacitások esetében a valószínűségi feltétel egyenlőséggel teljesül, tehát 0,8 a valószínűsége annak, hogy a három hónap egyikében sem lesz vízhiány. A teljes létesítési költség $99{,}3556 \cdot 10^6$ Ft.

IRODALOM

[1] DEÁK, I., „A többdimenziós tér halmazai valószínűségének kiszámítása normális eloszlás esetén", *Alk. Mat. Lapok* **2** (1976) 17—26.

[2] FIACCO, A. V. and MCCORMICK, G. P., *Nonlinear Programming: Sequential Unconstrained Minimization Techniques* (Wiley, New York, 1968).

[3] KOWALIK, J. and OSBORNE, M. R., *Methods for Unconstrained Optimization Problems* (Elsevier, New York, 1968).

[4] LOUCKS, D. P., "Stochastic methods for analysing river basin systems", *Research Project Technical Completion Report, OWRR Project No. C—1034* (Dept. of Water Resources Engineering, Cornell Univ., Ithaca, N. Y., 1969).

[5] MORAN, P. A. P., *The Theory of Storage* (Methuen, London, 1959).

[6] PRÉKOPA, A., "Logarithmic concave measures with application to stochastic programming", *Acta Sci. Math.* **35** (1971) 301—316.

[7] PRÉKOPA, A., "A class of stochastic programming decision problems", *Mathematische Operationsforschung und Statistik* **3** (1972) 349—354.

[8] PRÉKOPA, A., "Stochastic programming models for inventory control and water storage problems", in: *Colloquia Mathematica Societatis János Bolyai, 7. Inventory Control and Water Storage, Győr (Hungary), 1971* Ed. A. Prékopa (North Holland, 1973) 229—245.

[9] PRÉKOPA, A., "Logarithmic concave measures and related topics", *Proceedings of the International Conference on Stochastic Programming Oxford, 1974*, (Acad. Press, to appear).

[10] PRÉKOPA A. és SZÁNTAI T., „Egy új, többdimenziós gamma eloszlás és annak illesztése empirikus adatokhoz", Alkalmazott Matematikai Lapok 1 (1975) 299—318.

[11] RAPCSÁK T., Egy tározási modell számítástechnikai megoldása, Egyetemi doktori disszertáció, 1974.

(Beérkezett: 1977. augusztus 3.)

PRÉKOPA ANDRÁS ÉS RAPCSÁK TAMÁS
MTA SZTAKI, 1111 BUDAPEST, KENDE UTCA 13—17.
ZSUFFA ISTVÁN
ALSÓDUNAVÖLGYI VÍZÜGYI IGAZGATÓSÁG
6500 BAJA

A NEW METHOD FOR SERIALLY LINKED RESERVOIR SYSTEM DESIGN USING STOCHASTIC PROGRAMMING

A. PRÉKOPA, T. RAPCSÁK and I. ZSUFFA

A stochastic programming model formulated for a linearly linked reservoir system design is presented and a numerical solution method is proposed. The model was already formulated in [8] in a concise form. Here we give a more detailed explanation, apply SUMT for the solution of the optimization problem and present a numerical exemple. This is taken out from a larger model which arose in the first implementation of this reservoir system design method in the practice.

Alkalmazott Matematikai Lapok **2** *(1976) 203—217*

ÁRVÍZI TÁROZÓK MÉRETEZÉSE SZTOCHASZTIKUS PROGRAMOZÁSSAL

PRÉKOPA ANDRÁS ÉS SZÁNTAI TAMÁS
Budapest

A természetes vízrendszer matematikai modellje a gyökeres, irányított fa, melyben az élek irányítása megegyezik a vízfolyás irányával. Feltesszük, hogy bizonyos szelvényekben tározási lehetőség van és az ezekben építendő tározók feladata az árvíz felfogása lesz, mondjuk minden évben egy alkalommal. Feladatunk a tározók optimális kapacitásának meghatározása. Minimalizáljuk a teljes létesítési költség és egy büntető költség összegét egy megbízhatósági jellegű feltétel és a kapacitásokra tett alsó és felső korlátok mellett. A nyert nemlineáris programozási feladat megoldására Veinott támaszsík módszerét alkalmazzuk, ezt kombináljuk többdimenziós valószínűségeloszlásokkal kapcsolatos szimulációval. A feladat megoldását numerikus példákon illusztráljuk.

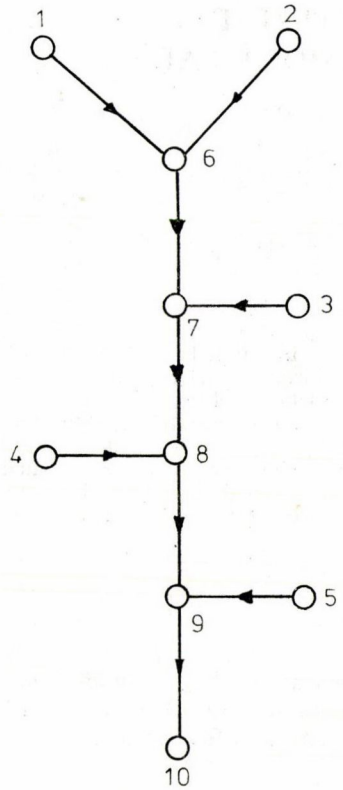
1. A feladat megfogalmazása

A feladatnak, amellyel foglalkozunk, több alkalmazása van. Legközvetlenebb az olyan tározórendszerek méretezésére vonatkozó alkalmazás, melyben a tározók rendeltetése az árvíz felfogása. Ezért a továbbiakban ennek a feladatnak a terminológiáját alkalmazzuk.

A gráfelméletben a *fa* általában olyan összefüggő, nem irányított gráfot jelent, melyben nincs körút. Ha egy csúcsot kiválasztunk és a fa gyökerének nevezzük, akkor a gráfnak gyökeres fa a neve.

Figyelembe véve a megfogalmazandó fizikai problémát, tárgyalásunk egyszerűsítése céljából beszélni fogunk gyökeres irányított fáról, mely a gyökeres fából oly módon származik, hogy az éleket irányítjuk. Az irányítást induktíve a következő módon adjuk meg: kiindulunk egy gyökérből, kiválasztjuk a szomszédos csúcsokat és a nyert élekhez hozzárendeljük azt az irányt, amely a gyökérből kifelé mutat; majd az említett, szomszédos élekből indulunk ki és ugyanezt tesszük, de a gyökeret kihagyjuk s. í.t. Ilyenformán olyan sémát kapunk, mint amilyet az 1. ábra ábrázol. Ez lesz számunkra a természetes vízrendszer topológiájának matematikai modellje.

Jelölje $n+1$ a gráf csúcsainak a számát és jelöljék e csúcsokat $a_1, \ldots, a_{n+1}$. Az a_i csúcsot az a_j csúcs elődjének nevezzük, ha van olyan irányított út, mely az a_i csúcsból az a_j csúcsba vezet. Ha ilyen út van, akkor ez nyilván egyértelműen meghatározott. Ha az út r élből áll, akkor azt mondjuk, hogy a_i az a_j r-edrendű elődje. Az előd nélküli csúcsot terminálisnak nevezzük. Az olyan élt is terminálisnak nevezzük, mely terminális csúcsból indul. Tegyük fel, hogy olyan jelölést választottunk, hogy $a_1, \ldots, a_m$ a terminális csúcsokat és a_{n+1} a fa gyökerét jelölik. Feltesszük, hogy az a_{n+1} csúcsba egyetlen él megy csupán és ez olyan folyószakaszt jelöl, melyen tározó létesíthető. Ha nem így volna, akkor a feladatot dekompanálhatnánk és



1. ábra
Vízrendszer topológiájának szemléltetése. Nyíl jelzi a vízfolyás irányát

legalább két, független tározórendszer méretezési problémát fogalmazhatnánk meg. Jelölje a_n azt a csúcsot, amellyel a_{n+1} össze van kötve.

Tegyük fel, hogy bizonyos, a gráfban élekkel jelölt folyószakaszokon tározási lehetőség van és a létesítendő tározók egyetlen feladata az időnként jelentkező árvíz felfogása lesz. Az árvízzel kapcsolatban feltesszük, hogy ez a terminális csúcsokban jelentkezik, véletlenszerűen, de ismerjük a véletlen vízmennyiségek együttes valószínűségeloszlását. Feltesszük, hogy a mederben elvezethető vízmennyiségeket már levontuk és az árvíz vízmennyiségein az eredeti vízmennyiségeknek csupán azt a részét értjük, melyet a tározókkal teljes egészében fel kell fognunk.

A tározók feltöltési politikáját illetően a következőkben állapodunk meg. Először a terminális élek tározóit töltjük fel. Ha ezáltal az árvizet még nem fogtuk fel teljesen, akkor továbbmegyünk és a túlfolyt vízmennyiségeket, továbbá azokat, amelyek tározó nélküli terminális élen jönnek, megkísérelük felfogni a legközelebbi tározókkal s. í. t.

Mindegyik terminális csúcshoz tartozik egy valószínűségi változó, mely a csúcsban jelentkező véletlen vízmennyiséget jelenti. Jelöljék ezeket $x_1, \ldots, x_m$. Az ismeretlen tározó méreteket a K_i szimbólumok jelölik. Az i indexet — kényelmi okból — oly módon választjuk meg, hogy ez megegyezzék annak a csúcsnak az indexével, amelyből a K_i tározót hordozó él kiindul. Értelmezzük ezek után rekurzive az $x_{m+1}, \ldots, x_{n+1}$ valószínűségi változókat a következő módon:

$$x_j = \sum_{i \in A_j} [x_i - \min(x_i, K_i)] + \sum_{i \in B_j} x_i, \quad j = m+1, \ldots, n+1, \tag{1.1}$$

ahol $A_j(B_j)$ az a_j csúcs ama elsőrendű elődeinek a halmaza, amelyek a_j-vel tározós (tározó nélküli) éllel vannak összekötve.

A tározórendszer akkor és csak akkor képes az árvíz felfogására, ha a következő feltétel teljesül:

$$x_{n+1} = 0, \tag{1.2}$$

vagy ami ugyanaz, fennáll az alábbi egyenlőtlenség

$$x_n \leqq K_n. \tag{1.3}$$

Vezessük be a következő jelölést

$$A = \bigcup_{j=m+1}^{n+1} A_j. \tag{1.4}$$

A K_i, $i \in A$ kapacitások meghatározására a következő sztochasztikus programozási feladatot fogalmazzuk meg

$$\text{minimalizálandó} \sum_{i \in A} [c_i(K_i) + E(\mu)]$$

$$P(x_n \leqq K_n) \geqq p, \tag{1.5}$$

feltéve, hogy

$$0 \leqq K_i \leqq V_i, \quad i \in A,$$

ahol p rögzített valószínűség, a gyakorlatban 1-hez közeli szám, a V_i, $i \in A$ számok a kapacitásokra vonatkozó számszerű felső korlátok, E a várható érték jele és μ az esetleges $x_n - K_n > 0$ rosszirányú eltérés büntetése. Ha lineáris büntető függvényt alkalmazunk, akkor

$$\mu = \begin{cases} q(x_n - K_n), & \text{ha} \quad x_n > K_n, \\ 0, & \text{egyébként,} \end{cases} \tag{1.6}$$

ahol q nemnegatív állandó.

Az (1.5) feladat variánsai is fontosak lehetnek a gyakorlat számára. Például előírhatjuk az árvízfelfogó képességet néhány részrendszerre is és így az egyetlen $x_n \leqq K_n$ egyenlőtlenség helyett több ilyen egyenlőtlenségünk lesz. Ezek után vagy egyetlen valószínűségi feltételt írunk elő ezeknek az egyenlőtlenségeknek a teljesülésére, vagy szeparált valószínűségi feltételekkel dolgozunk.

Ebben a tározórendszer méretezési modellben fontos szerepe van az $x_1, \ldots, x_m$ valószínűségi változók együttes eloszlásának. Egyszerű példák említhetők annak illusztrálására, hogy a tározók méretei szorosan függnek az együttes (és nemcsak az egyes valószínűségi változók) eloszlásától.

2. A tározórendszer méretezési modell matematikai tulajdonságai

Először felidézünk néhány olyan matematikai fogalmat, melyekre a későbbiekben szükségünk lesz.

Az R^k téren értelmezett nemnegatív értékű f függvényt logaritmikusan konkávnak (logkonkávnak) nevezzük, ha minden $\mathbf{x}, \mathbf{y} \in R^k$ és $0 < \lambda < 1$ esetén fennáll az alábbi egyenlőtlenség

$$f(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) \geqq [f(\mathbf{x})]^{\lambda} [f(\mathbf{y})]^{1-\lambda}. \tag{2.1}$$

Ha (2.1) helyett az

$$f(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) \geqq \min [f(\mathbf{x}), f(\mathbf{y})] \tag{2.2}$$

egyenlőtlenség érvényes, akkor azt mondjuk, hogy f kvázikonkáv. Ez utóbbi esetben f-nek nem kell nemnegatívnak lennie. (2.1) nyilván maga után vonja (2.2) teljesülését.

Az R^k tér mérhető halmazain értelmezett P valószínűségi mértéket logaritmikusan konkávnak (logkonkávnak) nevezzük, ha tetszőleges A, B konvex halmazok és $0 < \lambda < 1$ esetén fennáll a

$$P(\lambda A + (1-\lambda)B) \geqq [P(A)]^{\lambda} [P(B)]^{1-\lambda} \tag{2.3}$$

egyenlőtlenség. Ha (2.3) helyett a

$$P(\lambda A+(1-\lambda)B) \geqq \min [P(A), P(B)] \tag{2.4}$$

egyenlőtlenség érvényes, akkor azt mondjuk, hogy P kvázikonkáv valószínűségi mérték.

1. TÉTEL [5], [7]. Ha az R^k tér mérhető halmazain értelmezett P valószínűségi mértéket logkonkáv sűrűségfüggvény származtatja, akkor P logkonkáv mérték.

2. TÉTEL [2]. Ha az R^k tér mérhető részhalmazain értelmezett P valószínűségi mértéket olyan f sűrűségfüggvény származtatja, melyre fennáll, hogy $f^{-\frac{1}{k}}$ konvex függvény az egész téren, akkor P kvázikonkáv valószínűségi mérték.

3. TÉTEL [8]. Ha az $\mathbf{y} \in R^q$ valószínűségi vektorváltozó eloszlása logkonkáv (kvázikonkáv) valószínűségi mérték az R^q térben és $\mathbf{x}=\mathbf{A}\mathbf{y}+\mathbf{b}$, ahol $\mathbf{A}$ $k \times q$ méretű, konstans elemekkel bíró mátrix, $\mathbf{b} \in R^k$ pedig konstans vektor, akkor $\mathbf{x}$ eloszlása logkonkáv (kvázikonkáv) valószínűségi mérték az R^k térben.

4. TÉTEL [6]. Legyenek $g_1(\mathbf{K}, \mathbf{y}), \ldots, g_r(\mathbf{K}, \mathbf{y})$ a $\mathbf{K} \in R^k$, $\mathbf{y} \in R^q$ vektorokban foglalt összes komponens konkáv függvényei. Tegyük fel, hogy $\mathbf{y}$ valószínűségi vektorváltozó, melynek eloszlása logkonkáv (kvázikonkáv) valószínűségi mérték. Ekkor

$$h(\mathbf{K}) = P(g_i(\mathbf{K}, \mathbf{y}) \geqq 0,\ i = 1, \ldots, r) \tag{2.5}$$

logkonkáv (kvázikonkáv) függvénye a $\mathbf{K} \in R^k$ változónak.

A logkonkáv mértékekre példaként megemlítjük a normális eloszlást. Az R^k térben értelmezett nem elfajult normális eloszlás sűrűségfüggvénye a következő

$$f(\mathbf{z}) = \left[\frac{1}{(\det \mathbf{C})(2\pi)^n}\right]^{\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{z}-\mathbf{t})' \mathbf{C}^{-1}(\mathbf{z}-\mathbf{t})}, \quad \mathbf{z} \in R^k, \tag{2.6}$$

ahol $\mathbf{C}$ pozitív definit mátrix, a kapcsolódó valószínűségi vektorváltozó kovariancia mátrixa, $\mathbf{t}$ pedig a várható értékek vektora. Könnyű belátni, hogy f logkonkáv függvény. Ebből az 1. tétel szerint következik, hogy az f által generált valószínűségi mérték logkonkáv.

Második példánk egy speciális többdimenziós gamma eloszlás, melyet a [10] dolgozatban vezettünk be. Ez az

$$\mathbf{x} = \mathbf{A}\mathbf{y} \tag{2.7}$$

valószínűségi vektorváltozó komponenseinek együttes eloszlása, ahol $\mathbf{y}$ független, standard gamma eloszlású komponensekből alkotott valószínűségi vektorváltozó, $\mathbf{A}$ pedig olyan 0 és 1 elemekkel bíró mátrix, melynek oszlopait a 0, 1 komponensű összes különböző vektorok alkotják a zéró vektor kivételével.

Egy egydimenziós folytonos eloszlást standard gamma eloszlásnak nevezünk, ha sűrűségfüggvénye a következő alakú:

$$\frac{1}{\Gamma(\vartheta)} z^{\vartheta-1} e^{-z}, \quad \text{ha} \quad z > 0 \tag{2.8}$$

és zéró, ha $z \leqq 0$; ϑ pozitív állandó. Ha $\vartheta \geqq 1$, akkor ez a sűrűségfüggvény logkonkáv.

Ha **y** komkonenseinek paraméterei legalább 1-gyel egyenlők, akkor minthogy a komponensek együttes sűrűségfüggvénye egyenlő az egyes sűrűségfüggvények szorzatával, következik, hogy **y** eloszlása logkonkáv. A 3. tétel szerint ebből következik, hogy **x** eloszlása is logkonkáv.

Ha **y** bizonyos komponenseinek a paramétere kisebb, mint 1, akkor **x** eloszlása már nem biztos, hogy logkonkáv. Mégis az együttes eloszlásfüggvény mindig logkonkáv (pont) függvény. Ennél valamivel általánosabb állítást tartalmaz az alábbi

5. TÉTEL. Ha $\mathbf{A}_1$ nemnegatív elemekkel bíró mátrix és y független, standard gamma eloszlású komponensekkel bíró olyan valószínűségi vektorváltozó, melyben a komponensek száma megegyezik $\mathbf{A}_1$ oszlopainak számával, akkor $\mathbf{A}_1\mathbf{y}$ eloszlásfüggvénye, tehát a

$$P(\mathbf{A}_1\mathbf{y} \leqq \mathbf{z}) \tag{2.9}$$

függvény a **z** változó logkonkáv függvénye az egész téren.

Bizonyítás. Legyen y_i az **y** valószínűségi vektorváltozó egy komponense és jelölje ϑ_i az eloszlása paraméterét. Ha $\vartheta_i \geqq 1$, akkor az y_i valószínűségi változót meghagyjuk eredeti formájában. Ha azonban $\vartheta_i < 1$, akkor felírjuk a következő alakban

$$y_i = (y_i^{\vartheta_i})^{\frac{1}{\vartheta_i}} \tag{2.10}$$

és megjegyezzük, hogy $y_i^{\vartheta_i}$ az alábbi sűrűségfüggvénnyel bír

(2.11)

$$\frac{d}{dz} P(y_i^{\vartheta_i} < z) = \frac{d}{dz} P(y_i < z^{\frac{1}{\vartheta_i}}) = \frac{d}{dz} \int_0^{z^{1/\vartheta_i}} \frac{1}{\Gamma(\vartheta_i)} t^{\vartheta_i - 1} e^{-t} dt = \frac{1}{\Gamma(\vartheta_i + 1)} e^{-z^{1/\vartheta_i}},$$

ha $z > 0$ és zéró, ha $z \leqq 0$. Minthogy a (2.11) függvény logkonkáv sűrűségfüggvény és (2.10) szerint y_i konvex függvénye az $y_i^{\vartheta_i}$ valószínűségi változónak, a 4. tétel szerint a **z** változó

$$P(\mathbf{A}_1\mathbf{y} < \mathbf{z}) = P(\mathbf{z} - \mathbf{A}_1\mathbf{y} > 0)$$

függvénye logkonkáv az egész téren. Ezzel a tételt bebizonyítottuk.

6. TÉTEL. Ha $\mathbf{A}_1$ nemnegatív elemekkel bíró mátrix és **y** olyan pozitív komponensű valószínűségi vektorváltozó, mely komponensei logaritmusainak együttes eloszlása normális, továbbá **y**-nak ugyanannyi komponense van, mint amennyi oszlopa $\mathbf{A}_1$-nek, akkor az $\mathbf{A}_1\mathbf{y}$ valószínűségi vektorváltozó eloszlásfüggvénye logkonkáv (pont) függvény az egész téren.

Bizonyítás. A bizonyítás hasonló az 5. tétel bizonyításához. Csupán az

$$y_i = e^{\log y_i}$$

egyenlőséget kell felírnunk minden y_i komponensre és az állítás a 4. tétel alkalmazásával könnyen igazolható.

Véletlen szélsőértékek eloszlásának leírására kedvelt a *Gumbel eloszlás*. Tudomásunk szerint azonban még nem értelmeztek olyan többdimenziós *Gumbel eloszlást*, melyben a komponensek stochasztikusan függők. Ezért a *Gumbel eloszlás*nak a feladatunkra való alkalmazása csak speciális esetben lehetséges. A *Gumbel eloszlás* eloszlásfüggvénye a következő (egyváltozós függvény):

$$e^{-\lambda e^{-\mu z}}, \quad -\infty < z < \infty, \tag{2.12}$$

ahol $\lambda>0$, $\mu>0$ állandók. A deriváltat véve logkonkáv (pont) függvényt kapunk, amiből következik, hogy a *Gumbel eloszlás* logkonkáv.

Numerikus példáinkban a többdimenziós normális és a többdimenziós gamma eloszlást alkalmazzuk.

7. TÉTEL. Ha $j \geqq m+1$, akkor x_j konvex függvénye azoknak az $x_1, \ldots, x_m$ között elhelyezkedő változóknak, amelyek az a_j csúcs elődeihez tartoznak, továbbá azoknak a K_i változóknak, amelyeknek megfelelő él az a_j csúcsot egy elődjével, vagy a_j két elődjét köti össze. Speciálisan,

$$g(\mathbf{K}, \mathbf{x}) = K_n - x_n \tag{2.13}$$

konkáv függvénye az $\mathbf{x}$ és $\mathbf{K}$ változóknak, ahol $\mathbf{x}$ az $x_1, \ldots, x_m$ komponensekből, $\mathbf{K}$ pedig a K_i, $i \in A$ komponensekből alkotott vektor.

Bizonyítás. Megjegyezzük, hogy konvex függvények maximuma is konvex. Ha az (1.1) egyenlőségben elvégezzük az alábbi egyenlőségnek megfelelő helyettesítést

$$x_i - \min(x_i, K_i) = \max(0, x_i - K_i), \tag{2.14}$$

akkor olyan alakhoz jutunk, melyből állításunk indukcióval könnyen nyerhető.

8. TÉTEL. Ha $x_1, \ldots, x_m$ együttes eloszlása logkonkáv (kvázikonkáv) eloszlás, akkor

$$h(\mathbf{K}) = P(g(\mathbf{K}, \mathbf{x}) \geqq 0) \tag{2.15}$$

a $\mathbf{K}$ változó logkonkáv (kvázikonkáv) függvénye az egész téren.

Bizonyítás. Tételünk a 4. és a 7. tétel közvetlen következménye.

9. TÉTEL. Ha $\mathbf{x}$ eloszlása nem-elfajult normális eloszlás, akkor a $h(\mathbf{K})$ függvény az egész téren folytonos gradienssel bír. Ha $\mathbf{x}$ a fent említett többdimenziós gamma eloszlással bír, tehát $\mathbf{x}=\mathbf{A}\mathbf{y}$, ahol $\mathbf{A}$ és $\mathbf{y}$ eleget tesznek az említett feltételeknek, akkor $h(\mathbf{K})$ gradiense folytonos a tér legfeljebb ama $\mathbf{K}$ vektorainak a kivételével, melyek legalább egy komponense zéró.

Bizonyítás. Az $x_n \leqq K_n$ feltétel ekvivalens módon megfogalmazható egy lineáris egyenlőtlenségrendszerrel, melyben az $x_1, \ldots, x_m$ valószínűségi változók bizonyos részletösszegei kisebb-egyenlők, mint a K_i, $i \in A$ bizonyos részletösszegei. Ezt az egyenlőtlenségrendszert oly módon nyerhetjük, hogy (1.1)-ben elvégezzük a (2.14) egyenlőségnek megfelelő helyettesítést, majd minden $\max(a, b)=c$ típusú egyenlőtlenséget fokozatosan felbontunk $a \leqq c$, $b \leqq c$ alakra. Az egyenlőtlenségrendszerben x komponenseinek részletösszegeit mindenütt bal oldalra írva, e bal oldalak

együtt **x** lineáris transzformáltját alkotják. Jelölje **B** ennek mátrixát. Ennek sorai lineárisan független vektorok.

Tekintsük előbb a normális eloszlás esetét. Minthogy **x** eloszlása nem-elfajult, következik, hogy **Bx** bármely két komponense lineárisan független, azaz korrelációs együtthatójuk abszolút értéke 1-nél kisebb. Ebből következik, hogy **Bx** eloszlásfüggvényének a gradiense folytonos az egész téren (a parciális deriváltak feltételes eloszlásokkal kifejezhetők; a folytonosság e formulákból olvasható le.) Minthogy a (2.15) valószínűség **Bx** együttes eloszlásfüggvényéből oly módon nyerhető, hogy az utóbbi argumentumai helyére **K** komponenseinek részletösszegei kerülnek, állításunk a normális eloszlás esetére bizonyítást nyert.

A többdimenziós gamma eloszlás esetében hasonló módon járunk el. Vezessük be a $\mathbf{v}=\mathbf{Bx}$ jelölést. Jelölje továbbá s a **v** valószínűségi vektorváltozó komponenseinek a számát. A valószínűségelméletből ismeretes, hogy

$$\frac{d}{dz_1}P(v_1 < z_1, \ldots, v_s < z_s) = P(v_2 < z_2, \ldots, v_s < z_s | v_1 = z_1)g_1(z_1),$$

ahol $g_1(z_1)$ a v_1 valószínűségi változó sűrűségfüggvénye. Ez az utóbbi legfeljebb a $z_1=0$ esetben nem folytonos. A $g_1(z_1)$ mellett álló feltételes eloszlás a $z_1, z_2, \ldots, z_s$ változók folytonos függvénye. Ezt az állítást a [10] dolgozat (4.2) egyenlősége levezetéséhez alkalmazott gondolatmenethez hasonló módon láthatjuk be. Ismét figyelembe kell vennünk, hogy a (2.15) valószínűséget **Bx** eloszlásfüggvényéből oly módon származtatjuk, hogy az utóbbi argumentumaiba **K** komponenseinek alkalmas részletösszegeit helyettesítjük. Ha **K** mindegyik komponense pozitív, akkor e részletösszegek is pozitívak. Ebből következik, hogy a (2.15) valószínűség gradiense létezik és folytonos pozitív komponensű **K** esetén. Ezzel a tételt bebizonyítottuk.

10. TÉTEL. Az (1.15) feladat célfüggvényében levő $E(\mu)$ büntető tag a **K** változó konvex függvénye tetszőleges eloszlású **x** esetén. Ha **x** eloszlása nem-elfajult normális eloszlás, akkor $E(\mu)$ az egész téren folytonos gradienssel bír. Ha **x** a fent említett többdimenziós gamma eloszlású, akkor $E(\mu)$ folytonos gradienssel bír a pozitív komponensű **K** vektorok esetén.

Bizonyítás. A tétel a [9] dolgozat 3.1 tétele alapján könnyen belátható, a részletes bizonyítást mellőzzük.

3. Az (1.5) feladat megoldása

Az (1.5) feladat megoldására VEINOTT *támaszsík módszerét* [11] alkalmazzuk. Először röviden ismertetjük a módszert, majd megmutatjuk, hogyan alkalmazható feladatunk megoldására.

A következő nemlineáris programozási problémával foglalkozunk:

(3.1) minimalizálandó $h_0(\mathbf{x})$

feltéve, hogy $h_i(\mathbf{x}) \geqq 0, \quad i = 1, \ldots, m.$

Tegyük fel, hogy fennállnak a következő feltételek:

1. *Feltétel.* Létezik olyan C^1 korlátos konvex poliéder, mellyel fennáll az alábbi reláció

$$\{\mathbf{x}|h_i(\mathbf{x}) \geqq 0,\ i = 1, \ldots, m\} \subset C^1. \tag{3.2}$$

2. *Feltétel.* A $-h_0, h_1, \ldots, h_m$ függvények kvázikonkávak és C^1-en folytonos gradienssel bírnak.

3. *Feltétel.* Van olyan $\mathbf{z}^1$, melyre fennáll, hogy $h_i(\mathbf{z}^1)>0$, $i=1, \ldots, m$. Az eljárás két fázisból áll.

I. Fázis. Keresünk olyan $\mathbf{z}^1$ vektort, mely eleget tesz a 3. feltételnek.

II. Fázis. Egymás utáni iterációkat hajtunk végre; az r-edik iteráció az alábbi két lépésből áll.

1. *Lépés.* Megoldjuk a következő feladatot

$$\begin{aligned} &\text{minimalizálandó } h_0(\mathbf{x}) \\ &\text{feltéve, hogy } \quad \mathbf{x} \in K^r, \end{aligned} \tag{3.3}$$

ahol C^r korlátos konvex poliéder. Legyen $\mathbf{x}^r$ e feladat egy optimális megoldása. Ha $h_i(\mathbf{x}^r) \geqq 0$, $i=1, \ldots, m$, akkor $\mathbf{x}^r$ a (3.1) feladat optimális megoldása. Ha ez nem teljesül, következik a 2. lépés.

2. *Lépés.* Legyen λ^r az a legnagyobb λ $(0 \leqq \lambda \leqq 1)$ melyre teljesülnek az alábbi egyenlőtlenségek:

$$h_i(\mathbf{z}^1 + \lambda(\mathbf{x}^r - \mathbf{z}^1)) \geqq 0, \quad i = 1, \ldots, m.$$

Ezt az egydimenziós feladatot megoldhatjuk pl. a *Fibonacci módszer*rel. Vezessük be a következő vektort

$$\mathbf{y}^r = \mathbf{z}^1 + \lambda^r(\mathbf{x}^r - \mathbf{z}^1). \tag{3.4}$$

Ha $h_0(\mathbf{y}^r) - h_0(\mathbf{x}^r) \leqq \varepsilon$, akkor $\mathbf{y}^r$ a (3.1) feladat közelítő megoldása, ahol ε általunk választott kis szám. Ha ez nem teljesül, akkor választunk olyan i_r indexet, melyre $h_{i_r}(\mathbf{y}_r)=0$. Megalkotjuk a

$$C^{r+1} = \{\mathbf{x}|\mathbf{x} \in C^r,\ \nabla h_{i_r}(\mathbf{y}^r)(\mathbf{x} - \mathbf{y}^r) \geqq 0\} \tag{3.5}$$

korlátos konvex poliédert és következik az 1. lépés r helyett $r+1$ alkalmazásával. A 2. ábra a fenti eljárás egy iterációját szemlélteti.

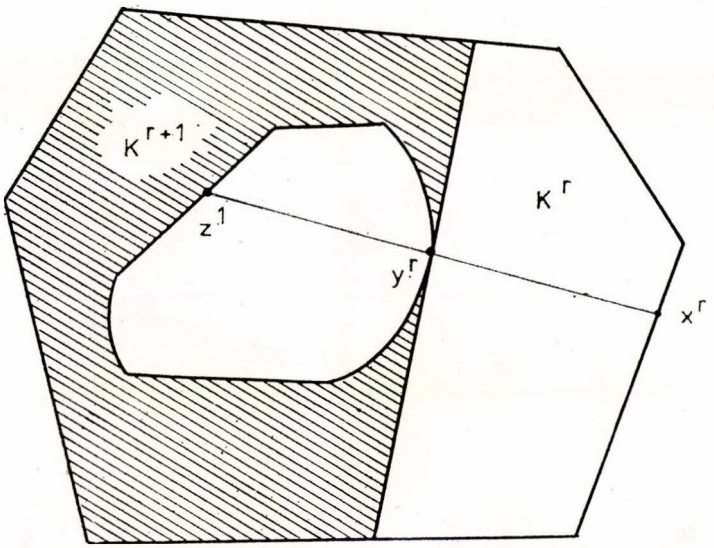
VEINOTT megmutatta [11], hogy ha a rögzített $\mathbf{z}^1$ vektor helyett az egyes iterációkban a

$$\mathbf{z}^{r+1} = \mathbf{z}^r + \beta(\mathbf{y}^r - \mathbf{z}^r), \quad r = 1, 2, \ldots \tag{3.6}$$

sorozatot használjuk, ahol $0<\beta<1$ rögzített szám akkor módszere ZOUTENDIJK *módosított megengedett irányok módszeré*re redukálódik.

Az (1.5) feladat megoldásakor a C^1 halmazt a következőképpen választjuk meg:

$$C^1 = \{\mathbf{K}|0 \leqq K_i \leqq V_i,\ i \in A\}.$$



2. ábra
Veinott támaszsík módszere egy iterációjának szemléltetése

Ellenőrizzük, hogy fennáll-e a $P(x_n \leqq K_n) > p$ egyenlőtlenség a $K_i = V_i$, $i \in A$ esetben; feltesszük, hogy ez a helyzet. E feltétel teljesülése maga után vonja a 3. feltétel teljesülését és a $K_i = V_i$, $i \in A$ komponensekből alkotott vektor megfelel a $\mathbf{z}^1$ vektor gyanánt. Csupán a II. fázist kell tehát végrehajtanunk. A belső pontot először a (3.6) relációnak megfelelően választottuk meg a $\beta = 0{,}5$ szorzóval. Minthogy azonban a $\mathbf{z}^r$, $r = 1, 2, \ldots$ sorozat igen gyorsan a valószínűségi feltétel által meghatározott halmaz egy határpontja felé tartott mielőtt az optimális megoldás jó közelítését megkaptuk volna, ezután a $\mathbf{z}$ vektorokat az alábbi szabály szerint választottuk

$$\mathbf{z}^{r+1} = \mathbf{z}^r + \frac{1}{r+1}(\mathbf{y}^r - \mathbf{z}^r), \quad r = 1, 2, \ldots . \tag{3.7}$$

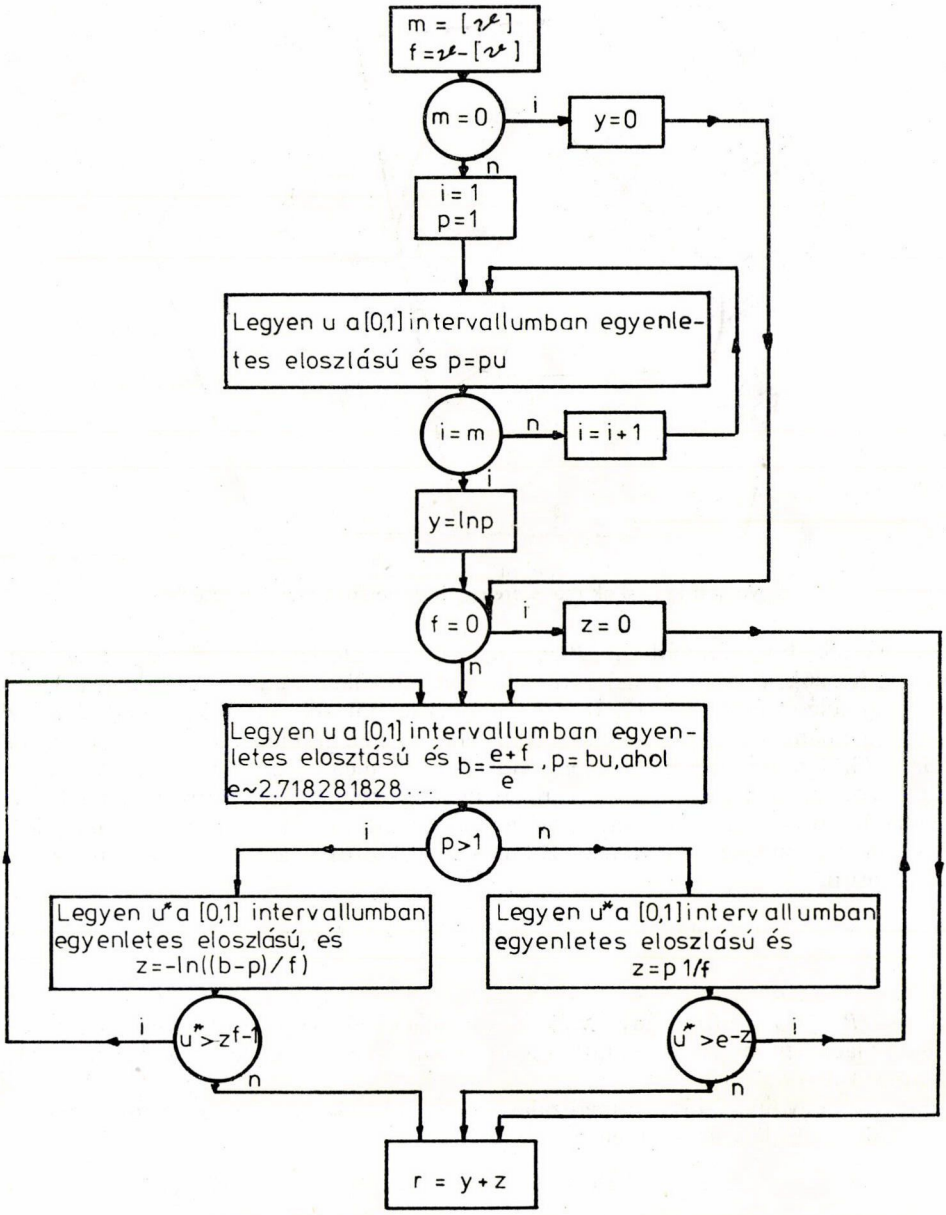
Ezáltal a módszer megjavult.

A $P(x_n \leqq K_n)$ függvény értékeit szimulációval határoztuk meg. Minthogy a szimuláció numerikus pontatlanságot eredményez, a következő óvintézkedést alkalmaztuk.

a) Az egydimenziós optimalizálás során olyan λ számot kerestünk, melyre az alábbi egyenlőtlenség érvényes

$$P(x_n \leqq K_n) - p \geqq -0{,}01$$

és a keresési eljárást leállítottuk 0,005, vagy ennél kisebb hiba esetén. Ezáltal az eljárás valamivel lassúbb lett, de megakadályoztuk, hogy az új vágás az (1.5) feladat megengedett pontjaiból levágjon. Különösen a gradiens pontatlan kiszámítása okoz ilyen problémát. Az egydimenziós optimalizálás egyszerű intervallum biszekció volt.



3. ábra
A standard gamma eloszlású valószínűségi változó generálásának folyamatábrája.
Ahrens és Dieter módszere

b) A $P(x_n \leqq K_n)$ függvény parciális deriváltjait a $K_i+0{,}1$ és a $K_i-0{,}1$ értékekhez tartozó függvényértékek szimulációs úton való meghatározásával számoltuk minden $i \in A$ esetén és ugyanezeket a véletlen számokat használtuk a gradiens meghatározására.

A függvényértékek meghatározására 1000, a valószínűségi feltételi függvény parciális deriváltjainak meghatározására 2000 elemű mintát alkalmaztunk.

A többdimenziós normális eloszlás esetében a véletlen számokat a SCICON Ltd. UNIVAC 1108 számítógépénél meglevő igen gyors algoritmussal generáltuk. Az algoritmust MARSAGLIA és BRAY adta meg [3]. Minden egyéb számolás is az előbb említett gépen történt. A gamma eloszlású véletlen számokat AHRENS és DIETER módszerével (*GT* algoritmus lásd [1]) generáltuk. Ennek folyamatábrája a 3. ábrán látható. Minden program FORTRAN nyelven íródott kivéve az egyenletes eloszlású véletlen számok generálására vonatkozó programot. Ez utóbbi assembly nyelven készült.

4. Numerikus példák

Tekintsük a 4. ábrán látható vízrendszer topológiát, ahol a lehetséges tározási helyeket is feltüntettük. Az $x_1, \ldots, x_{10}$ változó közül x_1, x_2, x_3, x_4, x_5 a terminális pontokon jelentkező vízmennyiségek. A többiek ezek segítségével a következő módon fejezhetők ki:

$$x_6 = x_1 - \min(x_1, K_1) + x_2 - \min(x_2, K_2),$$

$$x_7 = x_3 - \min(x_3, K_3) + x_6,$$

(4.1)

$$x_8 = x_4 + x_7,$$

$$x_9 = x_8 - \min(x_8, K_8) + x_5,$$

$$x_{10} = x_9 - \min(x_9, K_9).$$

Eszerint az (1.3) reláció a terminális változókkal és a kapacitásokkal a következőképpen írható fel:

$$x_9 = x_1 - \min(x_1, K_1) + x_2 - \min(x_2, K_2) +$$

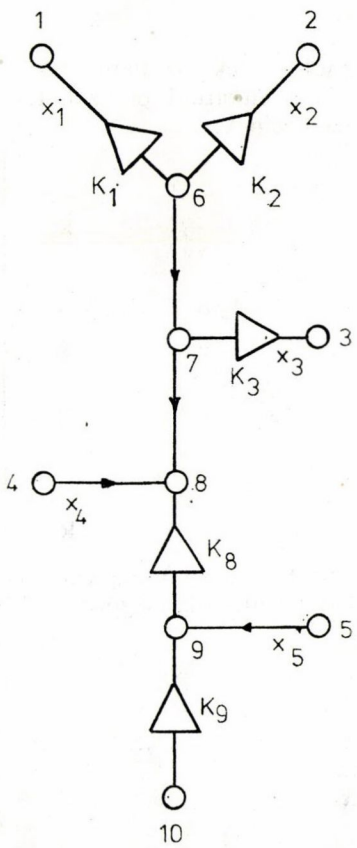
$$+ x_3 - \min(x_3, K_3) + x_4 - \min[x_1 - \min(x_1, K_1) +$$

(4.2)

$$+ x_2 - \min(x_2, K_2) + x_3 - \min(x_3, K_3) +$$

$$+ x_4, K_8] + x_5 \leqq K_9.$$

Nem fogunk büntető tagot alkalmazni, vagy ami ugyanaz, a büntető faktorokat zérónak választjuk.



4. ábra
Vízrendszer topológiájának és a lehetséges tározási helyeknek a szemléltetése. A vízfolyás iránya csak a tározó nélküli éleken van jelölve

Feladatunk a következő

$$\text{minimalizálandó } (0{,}4K_1+0{,}5K_2+0{,}6K_3+1{,}2K_8+1{,}8K_9)$$

feltéve, hogy

$$\begin{aligned} &P(x_9 \leqq K_9) \geqq p, \\ &0 \leqq K_1 \leqq 1, \\ &0 \leqq K_2 \leqq 1, \\ &0 \leqq K_3 \leqq 1, \\ &0 \leqq K_8 \leqq 2, \\ &0 \leqq K_9 \leqq 3, \end{aligned} \tag{4.3}$$

ahol p értéke az alábbi számpéldákban 0,8 illetve 0,9 lesz. Összesen 10 numerikus példát mutatunk be az 1. táblázatban. E példákban az alábbi korreláció mátrixok szerepelnek:

$$\mathbf{R}_1 = \begin{pmatrix} 1{,}0 & 0{,}0 & 0{,}6 & 0{,}4 & 0{,}0 \\ 0{,}0 & 1{,}0 & 0{,}5 & 0{,}3 & 0{,}3 \\ 0{,}6 & 0{,}5 & 1{,}0 & 0{,}7 & 0{,}6 \\ 0{,}4 & 0{,}3 & 0{,}7 & 1{,}0 & 0{,}4 \\ 0{,}0 & 0{,}3 & 0{,}6 & 0{,}4 & 1{,}0 \end{pmatrix},$$

$$\mathbf{R}_2 = \begin{pmatrix} 1{,}0 & -0{,}5 & 0{,}0 & 0{,}3 & -0{,}5 \\ -0{,}5 & 1{,}0 & -0{,}8 & 0{,}0 & 0{,}2 \\ 0{,}0 & -0{,}8 & 1{,}0 & 0{,}0 & 0{,}3 \\ 0{,}3 & 0{,}0 & 0{,}0 & 1{,}0 & 0{,}0 \\ -0{,}5 & 0{,}2 & 0{,}3 & 0{,}0 & 1{,}0 \end{pmatrix},$$

$$\mathbf{R}_3 = \mathbf{E},$$

ahol $\mathbf{E}$ az 5×5-ös egységmátrix. A várható értékek és a szórások valamennyi példában ugyanazok, mégpedig a következők (valamely egységben kifejezve):

	Várható értékek	Szórások
x_1	0,8	0,2
x_2	1,5	0,3
x_3	1,2	0,6
x_4	0,5	0,4
x_5	0,7	0,3

A többdimenziós gamma eloszlással kapcsolatban a [10] dolgozatban közölt előállítási technikát alkalmazzuk. Ez esetben csupán az $\mathbf{R}_1$ korreláció mátrix jön számításba, minthogy $\mathbf{R}_2$ negatív elemeket is tartalmaz, az $\mathbf{R}_3$ korreláció mátrix esete pedig ebből a szempontból triviális.

Az $\mathbf{R}_1$ korreláció mátrix esetében az alábbi előállításhoz jutunk:

$$x_1 = \frac{1}{20}(y_1+y_2+y_3 \qquad),$$

$$x_2 = \frac{3}{50}(\qquad +y_4+y_5+y_6+y_7 \qquad),$$

$$x_3 = \frac{3}{10}(y_1 \qquad +y_4+y_5 \qquad +y_8+y_9+y_{10}+y_{11} \qquad),$$

$$x_4 = \frac{8}{25}(\quad y_2 \qquad +y_6 \quad +y_8+y_9 \qquad +y_{12} \quad),$$

$$x_5 = \frac{9}{70}(\qquad y_4 \qquad +y_8 \quad +y_{10} \qquad +y_{13}),$$

ahol $y_1, \ldots, y_{13}$ független, standard gamma eloszlású valószínűségi változók a következő paraméterekkel

$$\begin{aligned} \vartheta_1 &= 0{,}5760 & \vartheta_8 &= 0{,}1400 \\ \vartheta_2 &= 0{,}1600 & \vartheta_9 &= 0{,}2800 \\ \vartheta_3 &= 15{,}2640 & \vartheta_{10} &= 0{,}0498 \\ \vartheta_4 &= 0{,}3150 & \vartheta_{11} &= 2{,}0550 \\ \vartheta_5 &= 0{,}5850 & \vartheta_{12} &= 0{,}7575 \\ \vartheta_6 &= 0{,}2250 & \vartheta_{13} &= 4{,}9404 \\ \vartheta_7 &= 23{,}8750 \end{aligned}$$

Teljesség kedvéért megadjuk az $x_9 \leqq K_9$ egyenlőtlenségnek azt az ekvivalens alakját, amely x_1, x_2, x_3, x_4, x_5 illetve K_1, K_2, K_3, K_8, K_9 részletösszegeit tartalmazza:

$$\begin{aligned} x_5 &\leqq K_9 \\ x_4 &\leqq K_8+K_9 \\ x_1+x_4 &\leqq K_1+K_8+K_9 \\ x_2+x_4 &\leqq K_2+K_8+K_9 \\ x_3+x_4 &\leqq K_3+K_8+K_9 \\ x_1+x_2+x_4 &\leqq K_1+K_2+K_8+K_9 \\ x_1+x_3+x_4 &\leqq K_1+K_3+K_8+K_9 \\ x_2+x_3+x_4 &\leqq K_2+K_3+K_8+K_9 \\ x_1+x_2+x_3+x_4 &\leqq K_1+K_2+K_3+K_8+K_9. \end{aligned}$$

Az optimális megoldásokat, az optimum értékeket és a számolási időket az 1. táblázatban adjuk meg.

1. TÁBLÁZAT

Numerikus eredmények

Az eloszlás típusa	Korreláció mátrix	Valószínűségi szint	K_1	K_2	K_3	K_8	K_9	Célfüggvény	Számolási idő
TÖBBDIMENZIÓS GAMMA	R_1	$p=0,8$	0,806735	1	1	1,355526	1,412243	5,591362	00:52:657
		$p=0,9$	0,751221	1	1	1,976085	1,398309	6,288746	00:35:688
	R_2	$p=0,8$	1	1	1	1,538713	1,193029	5,493909	00:16:785
		$p=0,9$	1	1	1	1,267790	1,848037	6,347815	00:11:343
TÖBBDIMENZIÓS NORMÁLIS	R_1	$p=0,8$	0,795523	1	1	1,590584	1,382698	5,815766	01:03:444
		$p=0,9$	0,997587	1	1	1,884778	1,524309	6,504525	00:25:126
	R_2	$p=0,8$	0,906312	1	1	1,350561	1,371008	5,551011	00:58:078
		$p=0,9$	0,833385	1	1	1,238889	1,830198	6,214377	00:51:426
	R_3	$p=0,8$	1	1	1	1,225805	1,430874	5,546541	00:43:461
		$p=0,9$	1	1	1	1,649903	1,373814	5,952749	00:57:478

percek
másodpercek
10^{-3} másodpercek

5. Köszönetnyilvánítás

A szerzők köszönetüket fejezik ki MARTIN BEALE professzornak, aki volt szíves felajánlani a SCICON Ltd. UNIVAC 1108 számítógépének a használatát.

IRODALOM

[1] AHRENS, J. H. and DIETER, U., "Computer methods for sampling from gamma, beta, Poisson and binomial distributions", *Computing* **12** (1974) 223—246.
[2] BORELL, C., "Convex set functions in *d*-space", *Periodica Mathematica Hungarica* **6** (1975) 111—136.
[3] MARSAGLIA, G. and BRAY, T. A., "A convenient method for generating normal variables", *SIAM Rev.* **6** (1964) 260—264.
[4] ORE, O., "Theory of graphs", *American Mathematical Society Coll. Publ.* **38** (1962).
[5] PRÉKOPA, A., "Logarithmic concave measures with application to stochastic programming", *Acta Scientiarum Mathematicarum* **32** (1971) 301—316.
[6] PRÉKOPA, A., "A class of stochastic programming decision problems", *Mathematische Operationsforschung und Statistik* **3** (1972) 349—354.
[7] PRÉKOPA, A., "On logarithmic concave measures and functions", *Acta Scientiarum Mathematicarum* **34** (1973) 335—343.
[8] PRÉKOPA, A., "Stochastic programming models for inventory control and reservoir system design", in: *Inventory Control and Water Storage, Coll. Math. Soc. J. Bolyai 7*, ed. A. Prékopa (North Holland Publ. Comp. Amsterdam—London, 1973), 229—245.

[9] PRÉKOPA, A., "Contributions to the theory of stochastic programming", *Mathematical Programming* **4** (1973) 202—221.
[10] PRÉKOPA, A. és SZÁNTAI, T., „Egy új, többdimenziós gamma eloszlás és annak illesztése empirikus adatokhoz", *Alkalmazott Matematikai Lapok* **1** (1975) 299—318.
[11] VEINOTT, A. F., "The supporting hyperplane method for unimodal programming", *Operations Research* **15** (1967) 147—152.

(Beérkezett: 1977. augusztus 3.)

PRÉKOPA ANDRÁS ÉS SZÁNTAI TAMÁS
BME GÉPÉSZMÉRNÖKI KAR MATEMATIKA TANSZÉK
1521 BUDAPEST XI. STOCZEK U. H ÉPÜLET, IV EM.

FLOOD CONTROL RESERVOIR SYSTEM DESIGN USING STOCHASTIC PROGRAMMING

A. PRÉKOPA and T. SZÁNTAI

Mathematically a natural river system is a rooted directed tree where the orientations of the edges coincide with the directions of the streamflows. Assume that in some of the river valleys it is possible to build reservoirs the purpose of which will be to retain the flood, once a year, say. The problem is to find optimal reservoir capacities by minimizing total building cost eventually plus a pentalty, where a reliability type constraint, further lower and upper bounds for the capacities are prescribed. The solution of the obtained nonlinear programming problem is based on the supporting hyperplane method of Veinott combined with simulation of multivariate probability distributions. Numerical illustrations are given.

Alkalmazott Matematikai Lapok **2** *(1976) 219—232*

KÉT BETEGSÉG—OSZTÁLY MEGKÜLÖNBÖZTETÉSÉRE SZOLGÁLÓ MATEMATIKAI MÓDSZEREK ALKALMAZÁSA A KORASZÜLÖTTEK KOPONYAŰRI VÉRZÉSE OKAINAK VIZSGÁLATÁRA[1]

SRAJBER BENEDEK, SEBŐK JÁNOS, FRITZ JÓZSEF, PAKSY ANDRÁS, KISZEL JÁNOS

Budapest

A cikk két betegség kategória megkülönböztetésére szolgáló matematikai módszereket és azok számítógépes realizációját mutatja be. A módszerek összehasonlítását, a számítógépes tapasztalatokat és az eredmények értékelését konkrét alkalmazáson, a koraszülöttek megfigyelt adataiból (egyedenként 31 befolyásoló tényező) kiindulva, a koponyaűri vérzést kiváltó okok vizsgálatán keresztül tárgyalja.

1. Két betegség-osztály megkülönböztetésére alkalmazott módszerek

A feladat megfogalmazása

Induljunk ki abból, hogy megfigyeléseink két betegség-kategóriára (osztályra) vonatkoznak. Jelöljük az egyik típusba tartozó betegek halmazát C_1-gyel, a másikat C_2-vel. Tételezzük fel, hogy a C_1, vagy C_2 osztályba sorolható egyedek jellemzésére — szakmai meggondolások alapján — összeállítottunk összesen m számú megfigyelendő tényezőt (tünetek és egyéb befolyásoló tényezők). Legyenek ezek: $\xi_1, \xi_2, \ldots, \xi_m$. Egy egyed jellemzésére az $\mathbf{x}=(\xi_1, \xi_2, \ldots, \xi_m)$ vektort fogjuk használni. Legyen a megfigyelt egyedek száma n, a hozzájuk tartozó m-komponensű vektorokat pedig jelölje rendre:

$$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n.$$

Tetszőleges $\mathbf{x}_j$ $(j=1, 2, \ldots, n)$ megfigyelésnek a C_1, ill. C_2 osztályhoz tartozását is jelezhetjük egy további $(m+1)$-edik komponens (ξ_{m+1}) bevezetésével a következő módon. Legyen

$$\varrho_j = \begin{cases} 1, & \text{ha az egyed } C_1\text{-hez tartozik,} \\ -1, & \text{ha az egyed } C_2\text{-höz tartozik;} \end{cases}$$

és

$$\mathbf{y}_j = (\mathbf{x}_j, \varrho_j), \quad j = 1, 2, \ldots, n \tag{1.1}$$

pedig a származtatott vektor.

Ezek után azt mondhatjuk, hogy vizsgálataink kiinduló alapját az n db vektorból álló (1.1) „tananyag" (minta) képezi. Az $\mathbf{y}_j$ vektorokról feltételezzük, hogy

[1] Jelen dolgozat az MTA 1975. évi pályázatán elnöki jutalomban részesült pályamunka matematikai és számítástechnikai vonatkozásait részletezi. Az eredmények bővebb orvosi értékelése és az orvosi szakirodalmi hivatkozások az *Orvosi Hetilap* 1976. máj. 9-i számában találhatók.

teljesen függetlenek és ugyanolyan eloszlásúak. Feladatunk abban áll, hogy az (1.1) minta alapján, a lényeges komponensek kiemelésével jellemezzük a C_1 és C_2 osztályokat oly módon, hogy egy új egyedre vonatkozó $\mathbf{x}$ megfigyelés esetén közvetlenül dönteni tudjunk a megfelelő osztályba sorolásról.

A megoldás matematikai modellje

Valamely $\mathbf{x}$ megfigyelés osztályba sorolásához azt kellene tudnunk, hogy milyen valószínűséggel jön számításba az első, ill. második osztály. Az első ill. második osztályba tartozás feltételes valószínűségeit jelölje $P(\varrho=1|\mathbf{x})$, ill. $P(\varrho=-1|\mathbf{x})$, amelyeket aposteriori valószínűségeknek nevezünk. Ezek ismeretében a

$$D(\mathbf{x}) = P(\varrho = 1|\mathbf{x}) - P(\varrho = -1|\mathbf{x})$$

függvény előjele alapján dönthetnénk, mégpedig, ha az előjel pozitív $(D(\mathbf{x})>0)$, akkor a C_1 osztályba tartozás mellett, ha negatív $(D(\mathbf{x})<0)$, akkor a C_2 osztályba tartozás mellett [2]. Ilyen típusú döntési szabály elkészítése az alakfelismerés alábbi elve szerint történhet [7]. Először olyan tényezőket (tüneteket, tünetcsoportokat) választunk ki, amelyek esetenkénti fennállása az egyik, vagy a másik lehetőséget valószínűsíti. Matematikailag ez úgy fogalmazható, hogy a kiválasztott tényezőkhöz rendre hozzárendelünk olyan $\varphi_1, \varphi_2, \ldots, \varphi_m$ számokat, hogy $\varphi_i=1$ ill. $\varphi_i=-1$ aszerint, hogy az i-edik befolyásoló tényező (csoport) fennállása a C_1, illetve C_2 osztályt valószínűsíti. A φ_i értékek az $\mathbf{x}$ megfigyelés alapján esetenként számolhatók, tehát egy $\varphi_i=\varphi_i(\mathbf{x})$ függvény határozza meg azokat. Ezután a döntés legegyszerűbb módja az lenne, hogy képezzük a $\sum_{i=1}^{m} \varphi_i(\mathbf{x})$ összeget, és ha ez pozitív (vagyis a C_1-et valószínűsítő tényezők vannak túlsúlyban), akkor a C_1, ha negatív, akkor a C_2 mellett döntünk. Ennél lényegesen jobb eredményt kaphatunk, ha az egyes tényezőket különböző (a jelentőségüknek megfelelő) súlyokkal vesszük figyelembe, tehát a

$$D(\mathbf{x}, \mathbf{c}) = \sum_{i=1}^{m} \gamma_i \varphi_i(\mathbf{x}) \tag{1.2}$$

összeget képezzük, ahol $\mathbf{c}=(\gamma_1, \gamma_2, \ldots, \gamma_m)$ a súlyokból alkotott vektor. Ha (1.2) pozitív, akkor C_1, ha negatív, akkor C_2 mellett döntünk. Itt a γ_i súlyokat úgy kell meghatározni, hogy a $D(\mathbf{x}, \mathbf{c})$ döntési szabállyal adott diagnózis a lehető legmegbízhatóbb legyen, tehát a $D(\mathbf{x}, \mathbf{c})$ előjele lehetőleg megegyezzen a ϱ_j-vel. A diagnosztikai megbízhatóságot a $\frac{k}{n}$ hányados adja meg, ahol k a helyesen diagnosztizált esetek száma (azoké, ahol ϱ_j és $D(\mathbf{x}, \mathbf{c})$ előjele azonos), n pedig az összes esetek száma.

Ha a $\frac{k}{n}$ hányados elég nagy (kb. 0,7—0,9), akkor már mondhatjuk, hogy a $D(\mathbf{x}, \mathbf{c})$ döntési szabály a dolog lényegét ragadja meg, tehát a C_1 osztály szempontjából azok a tényezők a legfontosabbak, melyeknek a súlya (a $\mathbf{c}$-vektor megfelelő komponense) a legnagyobb.

Ha a két osztály tökéletesen elválasztható lenne, akkor $D(\mathbf{x}, \mathbf{c})>0$, ill. <0 teljesülne, ha $\mathbf{x}_j$ a C_1, ill. a C_2 osztályhoz tartozik, tehát

$$\varrho_j D(\mathbf{x}_j, \mathbf{c}) = \varrho_j \sum_{i=1}^{m} \gamma_i \varphi_i(\mathbf{x}_j) > 0, \quad j = 1, 2, \ldots, n,$$

vagyis a

$$\sum_{i=1}^{m} \gamma_i \varrho_j \varphi_i(\mathbf{x}_j) > 0, \quad j = 1, 2, \ldots, n \tag{1.3}$$

egyenlőtlenségi rendszernek volna megoldása. Ez a megoldás azonban nem mindig létezik. Helyette közelítő megoldást keresünk különböző veszteség függvények minimalizálásával.

Az

$$\mathbf{A} = \{a_{ij}\} = \{\varrho_j \varphi_i(\mathbf{x}_j)\} \tag{1.4}$$

jelöléssel az $\mathbf{Ac}>0$ feladatot kapjuk, amelynek közelítő megoldása különböző módszerekkel nyerhető. A diagnosztikai megbízhatóságot az optimális megoldásnál nyert $\mathbf{Ac}$ vektor pozitív komponensei számának (k) és az összes esetek számának (n) hányadosa szolgáltatja. Ennek a kritériumnak az optimalizálására diszkrét programozási módszerek használhatók, de az $\mathbf{A}$-mátrix méretei miatt ezek alkalmazására nem gondolhatunk.

Az (1.3) feladat megoldására adott alábbi módszereink egy-egy diagnosztikai megbízhatósághoz hasonló, de könnyebben kezelhető veszteségfüggvény minimalizálására szolgáltak [6].

A Kashyap—Ho-féle gradiens algoritmus

Vezessük be segédváltozóként a $\mathbf{b}=(\beta_1, \beta_2, \ldots, \beta_n)$ vektort és tegyük fel, hogy $\mathbf{b}$ minden komponense legalább 1 ($\beta_j \geqq 1$). Ily módon feladatunk (1.3) helyett az eltérések négyzetösszegének (norma-négyzet) minimalizálására redukálódik, amelyet tömören így jelölünk:

$$\|\mathbf{Ac}-\mathbf{b}\|^2 \to \text{minimum}. \tag{1.5}$$

Az algoritmus lényege, hogy kezdő $\mathbf{c}$ és $\mathbf{b}$ értékekből $(\mathbf{c}_0, \mathbf{b}_0)$ kiindulva iteratív eljárással jutunk el az (1.5) feladat optimális megoldásához [3, 4]. A k-adik iteráció lépései:

a1) $\varepsilon_k = \mathbf{Ac}_k - \mathbf{b}_k$

a2) $\mathbf{c}_{k+1} = \mathbf{c}_k + \mathbf{S}\gamma\mathbf{A}^*\varepsilon_k$, ahol $\mathbf{A}^*$ az $\mathbf{A}$ transzponáltja, $\mathbf{S}=(\mathbf{A}^*\mathbf{A})^{-1}$, $0<\gamma\leqq 3$, az $|\varepsilon_k|$ pedig az ε_k abszolút értéke;

a3) $\mathbf{b}_{k+1} = \mathbf{b}_k + |\varepsilon_k| + \varepsilon_k$

a4) Befejező kritérium: $\dfrac{\|\varepsilon_k - \varepsilon_{k-1}\|^2}{\|\varepsilon_2 - \varepsilon_1\|^2} < 10^{-3}$.

(Egy másik befejező kritérium alapjául az $\mathbf{Ac}$ negatív komponensei számának csökkenését vettük.)

Az $\mathbf{S}$ mátrix olyan, hogy az $\mathbf{R}=2\gamma\mathbf{S}-\gamma^2\mathbf{SA}^*\mathbf{AS}$ mátrix pozitív definit és csupán a konvergencia meggyorsítására szolgál. Kimutatható, hogy az eljárás konvergens és az (1.5) minimalizálásával a

$$h(\mathbf{c}) = E\left(\frac{1+\operatorname{sign}[1-D(\xi,\mathbf{c})\varrho]}{2}\cdot[1-D(\xi,\mathbf{c})\varrho]^2\right)$$

veszteségfüggvény minimumhelyére kapunk becslést. Algoritmusaink közül a *Kashyap—Ho*-féle konvergált a leggyorsabban.

Lineáris regressziós módszer [5]

Az (1.3) feladat közelítő megoldásaként (1.5) mellett a

$$\sum_{j=1}^{n}\left(\sum_{i=1}^{m}\gamma_i\varphi_i(\mathbf{x}_j)-\varrho_j\right)^2 \to \min.$$

megoldása is szóba jöhet. A ϱ_j-ket a $\varphi_i(\mathbf{x}_j)$-k lineáris kombinációjaként határozzuk meg és egyetlen lépésben nyerjük a γ_i súlyok $\mathbf{c}$ vektorát a többszörös lineáris regresszió számításánál megszokott $\mathbf{c}=\mathbf{SA}^*\mathbf{b}_0$ formula alapján, ahol a $\mathbf{b}_0$ vektor minden komponense 1. A lineáris regressziónál minimalizálandó négyzetösszeg tagjait $\varrho_j\varrho_j=1$-gyel szorozva, a $\|\mathbf{Ac}-\mathbf{b}\|^2\to\min$ átfogalmazást kapjuk. Ez a módszer a minta együttes normális eloszlása esetén pontos, de más esetekben is jól használható, egyszerű számolhatósága miatt.

Módosított Kashyap—Ho eljárás

A *Kashyap—Ho algoritmus* általunk *módosított* (egyszerűbb) *változatának* iterációs lépései:

I. $\mathbf{c}_{k+1}=\mathbf{SA}^*\mathbf{b}_k$

II. $\|\mathbf{Ac}_{k+1}-\mathbf{b}\|^2\to\min$. feladat megoldása:

$$\mathbf{b}_{k+1}=\begin{cases}\mathbf{Ac}_{k+1}, & \text{ha } \mathbf{Ac}_{k+1} \text{ megfelelő komponense nem negatív}\\ 0, & \text{különben}\end{cases}$$

(Az eljárás konvergenciája bizonyítható.)

Lineáris programozási módszer

Az (1.5) feladat helyett az $\mathbf{Ac}$ vektor negatív komponensei abszolút értékei összegének minimalizálására törekszünk. Ezt az alábbi módon érjük el [7, 8, 9].

Legyen $\mathbf{E}$ egy $n\times n$-es egységmátrix, a $\mathbf{b}=(\beta_1,\ldots,\beta_n)$ egy nem negatív változókból ($\beta_j\geqq 0$), az $\mathbf{0}=(0,\ldots,0)$ pedig a nulla komponensekből álló vektor és oldjuk meg az

$$\mathbf{Ac}+\mathbf{Eb}\geqq\mathbf{0}$$

$$-1\leqq\gamma_i\leqq 1,\quad \sum_{i=1}^{m}\gamma_i=1,\quad i=1,\ldots,m \tag{1.6}$$

$$\sum_{j=1}^{n}\beta_j\to\min.$$

lineáris programozási feladatot. („A γ_i-kre tett feltevés alapján a feltételrendszer jobb oldala különbözni fog nullától.) Optimális megoldás esetén a célfüggvény, $\sum_{j=1}^{m} \beta_j$ éppen az **Ac** negatív komponensei abszolút értékeinek az összegével lesz egyenlő, amelyről belátható, hogy az (1.2) szeparáló függvénynek egy jó közelítéséhez vezet. Pontosabban a

$$h_2(\mathbf{c}) = \frac{1}{2} E\left(D(\xi, \mathbf{c}) \cdot [\operatorname{sign} D(\xi, \mathbf{c}) - D^*(\xi)]\right)$$

minimumhelyének becsléséről van szó, ahol $D^*(\xi)$ a *Bayes-féle szeparáló függvény*t jelöli.

A lineáris programozási eljárás lényegesen gyorsabbá tehető, ha a γ_i-kre előzetes becslést adunk (lásd 2. szakasz d) pontja).

2. A mintaanyag megválasztása és a feldolgozás koncepciója

A KSH és az Egészségügyi Minisztérium által 1970-ben végzett országos felmérés adatlapjait használtuk fel munkánkban. Az élveszületett és a korai neonatalis időszakban (0—6 nap) elhalt 1000—2500 g súlyú koraszülöttekből két csoportot alkottunk. Az egyik csoportba 500 olyan eset került, akiknél a boncolás intracraniális vérzést igazolt, a másikba pedig 500 olyan eset, ahol a kórboncnok a koponyaüregben vérzést nem talált.

A feldolgozás során az 1000 g-nál kisebb súllyal születetteket nem vettük számításba, mert az éretlenség (1000 g alatti születési súly), s a velejáró rendkívüli sérülékenység hibás következtetésekre vezethet. Vizsgálatainkból kizártuk azokat az eseteket is, ahol az élettel össze nem egyeztethető fejlődési rendellenességet tapasztaltunk.

Vizsgálataink a koraszülöttek alábbi adataira (befolyásoló tényezők) vonatkoznak:

1. (Sorszám),
2. Koponyaűri vérzés,
3. Fejlettségi kor hetekben ($\leqq$24, 25—27, 28—31, 37—41, $\geqq$42),
4. Nem (fiú, lány),
5. Születési súly g-ban (1001—1500, 1501—2000, 2001—2500),
6. Fejvégű fekvés,
7. Medencevégű fekvés,
8. Harántfekvés,
9. A szülés lefolyása (rendes vagy szövődményes),
10. A szülés befejeződése spontán,
11. A szülés befejeződése császármetszés,
12. A szülés befejeződése vacuum extractio,
13. A szülés befejeződése fogó-műtét,
14. A szülés befejeződése egyéb (extractio, expressio),
15. Terhesség alatt keletkezett vesebetegség,
16. Toxaemia gravidarum,
17. Ecclampsia,

18. A szülőerők rendellenessége,
19. Pyelitis, pyelonephritis gravidarum,
20. Placenta praevia,
21. Korai lepényleválás,
22. Köldökzsinór-rendellenességek,
23. Méhnyak elégtelenség,
24. Idő előtti burokrepedés,
25. Hányadik terhesség (1, 2, $\geqq 3$),
26. A kórelőzményben élveszületettek száma (1, 2, $\geqq 3$),
27. A kórelőzményben halvaszületettek száma (1, 2, $\geqq 3$),
28. A kórelőzményben művi vetélések száma (1, 2, 3, $\geqq 4$),
29. A kórelőzményben spontán vetélések száma (1, 2, 3, $\geqq 4$),
30. Az anya életkora ($\leqq 20$, 21—25, 26—30, 31—35, 36—40, $\geqq 41$),
31. Az anya dohányzik-e?

A nagy mintaanyag feldolgozása, melyet magától értetődően számítógéppel végeztünk (másként el sem képzelhető), három részre tagolható:

A) Első lépésben az agyvérzés egyes tényezőkre vonatkozó relatív gyakoriságát számítottuk ki, majd *u-próbával* meghatároztuk a szignifikancia szinteket. A legkedvezőbben szignifikáns tényezőket kiválogattuk és a továbbiakban azokat elemeztük. (A nem lényegesnek talált tényezőket elhagytuk.)

B) A második lépésben két betegség-osztály szétválasztására általánosan alkalmazható matematikai módszerek *(Kashyap—Ho-féle gradiens algoritmus*, annak *egy módosított változata*, *lineáris regresszió*, *lineáris programozási módszer)* számítógépes realizációját hajtottuk végre. Konkrét alkalmazásként meghatároztuk az agyvérzést „lényegesen befolyásoló tényezőkhöz" tartozó súlyokat.

C) A harmadik lépésben az agyvérzést kiváltó okok között legnagyobb súllyal szereplő hat tényező különböző kombinációinak előfordulásához — az agyvérzés tünetkombinációkra vonatkozó feltételes relatív gyakoriságának kiszámításával — megbízhatósági intervallumokat határoztunk meg 95 százalékos megbízhatósági szint mellett.

A számítógépes realizáció a konkrét feladat tükrében

Az alábbi jelöléseket vezetjük be:

n: A vizsgált újszülöttek száma (esetünkben $n=1000$; 500 agyvérzésben és 500 nem agyvérzésben meghalt),

m: Az egyes újszülötteknél figyelembe vett adatok (befolyásoló tényezők) száma (esetünkben 56),

B_i: Esemény, amely valamely újszülötteknél az i-edik befolyásoló tényező fennállását jelenti ($i=1, 2, \ldots, m$),

n_i: A vizsgált újszülöttek között az i-edik tényező előfordulásának gyakorisága ($i=1, 2, \ldots, m$),

x_j: A j-edik újszülött adataiból álló vektor ($j=1, 2, \ldots, n$),

$y_{ji}=\begin{cases}1, \text{ ha a } j\text{-edik újszülöttnél az } i\text{-edik tényező előfordul és agyvérzés áll fenn,}\\ 0, \text{ különben.}\end{cases}$

Az ezres mintaanyag feldolgozásának lépései

a) Az agyvérzésnél szóba jöhető valamennyi tényező egyöntetű vizsgálata. A

$$\bar{P}(\varrho_j = 1 | B_i) = \frac{\sum_{j=1}^{n} y_{ji}}{n_i}, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n$$

feltételes relatív gyakoriságok kiszámítása után ellenőriztük azt a hipotézist, hogy az agyvérzés (ill. nem agyvérzés) egyes befolyásoló tényezőire vonatkozó feltételes valószínűség érték 1/2-del egyezik-e meg. Az adott feltételek mellett az

$$u = [\bar{P}(\varrho_j = +1 | B_i) - 1/2] 2\sqrt{n}$$

változó közelítőleg standard normális eloszlású; ebből kiindulva megállapítjuk a szignifikánsnak mondható tényezőket a

$$P(|u| \geqq u_\varepsilon) = 2[1 - \Phi(|u|)]$$

reláció alapján ($\Phi(x)$ a standard normális eloszlásfüggvény), majd kiválogatjuk az „erősen" szignifikáns tényezőket.

Az agyvérzés egyes tényezőkre vonatkozó feltételes relatív gyakoriságaiból és a megfelelő szignifikancia értékekből (1. táblázat) nyilvánvalóvá vált, hogy mely tényezőknek nincs jelentősége és mely tényezők lehetnek jelentősek a koponyaűri vérzés szempontjából. Az előbbieket és az 1. táblázatban felsorolt tényezők közül a köldökzsinór anomáliákat elhagytuk. Az utóbbiak olyan szembetűnő elváltozások voltak, amelyeket igen gyakran rávezettek az adatlapokra akkor is, amikor a köldökzsinór keringése nem károsodott, tehát az anomália csupán a köldökzsinór helyzetére vonatkozott. A placenta praeviát pedig annak ellenére, hogy 0,26 szignifikanciával fordul elő, fontosnak ítéltük az agyvérzés létrejötte szempontjából. Így a továbbiakban csupán az alábbi tényezők agyvérzésre vonatkozó befolyását tartottuk szem előtt:

1. toxaemia csoport (terhesség alatt kezdődött vesebetegség, praeeclampsia, eclampsia),
2. rendellenes lefolyású szülés,
3. placenta praevia,
4. a szülőerők rendellenessége,
5. harántfekvés, medencevégű fekvés,
6. szülés befejezése expressioval, extractioval,
7. pyelitis, pyelonephritis gravidarum,
8. születési súly,
9. kórelőzmény, spontán vetélések,
10. császármetszés,
11. az anya életkora,
12. az előzményben élveszületések,
13. hányadik terhesség,
14. fejlettségi kor,
15. az előzményben művi vetélések.

1. TÁBLÁZAT

A koraszülöttek koponyaűri vérzésének létrejöttében legfontosabbnak bizonyult tényezők

	Tényező	Esetszám	Agyvérzés gyakorisága	Szignifikancia p
1.	Toxaemia	121	0,63636	0,002
2.	Ecclampsia	13	0,84615	0,01
3.	Terhesség alatt keletkezett vesebetegség	17	0,82353	0,01
4.	Köldökzsinór rendellenességek	47	0,65957	0,03
5.	Születési súly 1001—1500 g	525	0,54676	0,04
6.	Születési súly 1501—2000 g	325	0,44615	0,05
7.	Méhnyak elégtelenség	43	0,65116	0,05
8.	Harántfekvés	55	0,61818	0,08
9.	Szülés befejezése vacuum extractioval	6	0,83333	0,10
10.	Szülőerők rendellenessége	100	0,58000	0,11
11.	Rendes lefolyású szülés	691	0,52822	0,14
12.	Az előzményben 2 művi vetélés	100	0,57000	0,16
13.	Az előzményben 4 művi vetélés	43	0,00465	0,17
14.	Az anya pyelitise, pyelonephritise	22	0,03636	0,20
15.	A szülés befejezése extractioval, expressioval	109	0,55963	0,21
16.	Az előzményben 3 spontán abortus	17	0,35294	0,23
17.	II. terhesség	225	0,53778	0,26
18.	Születési súly 2001—2500 g	145	0,45717	0,28
19.	Korai lepényleválás	84	0,44048	0,28
20.	Előzményben 1 spontán vetélés	176	0,46023	0,29
21.	Placenta praevia	94	0,55319	0,30
22.	III. terhesség	528	0,47719	0,30
23.	Az anya életkora 36—40 év között	161	0,45363	0,31
24.	Előzményben 2 élveszülés	123	0,45528	0,32
25.	Előzményben 2 halvaszülött	1	0,0000	0,32
26.	Szülés befejezése fogóműtéttel	1	1,0000	0,32
27.	Fejlettségi kor 42 hét	1	1,0000	0,32
28.	Gestációs idő 24 hét	44	0,43182	0,37
29.	Szülés befejezése császármetszéssel	136	0,46324	0,39
30.	Előzményben 3 művi vetélés	37	0,43243	0,41
31.	Gestációs idő 32—36 hét	397	0,51889	0,45
32.	Előzményben 1 művi vetélés	211	0,47393	0,45
33.	Gestációs idő 37—41 hét	53	0,45283	0,49

b) A $\varphi_i(\mathbf{x}_j)$ valós számok meghatározása. A kiemelt 15 befolyásoló tényező 1. táblázatban szereplő adatai alapján származtatjuk a φ_i-ket ($i=1, 2, \ldots, k$; esetünkben $k=16$) lényegében a következő módon:

$$\varphi_i = \begin{cases} 1, \text{ ha egy tényező fennáll és az agyvérzés irányában szignifikáns,} \\ -1, \text{ ha egy tényező fennáll és nem az agyvérzés irányában szignifikáns,} \\ 0, \text{ különben.} \end{cases}$$

c) Az **A** mátrix előállítása (1.4) alapján. (Az **A** mátrix méretét 1000×56-ról 1000×16-ra csökkentettük azzal, hogy azokat a tényezőket elhagytuk, amelyek az agyvérzésre nézve nem voltak jelentős befolyással.)

d) A „c" kezdőértékének becslése: $\mathbf{c}_0$ meghatározása. Legyen

$$p_i = P(\varphi_i = 1 | \varrho_j = 1)$$
$$\bar{p}_i = P(\varphi_i = 1 | \varrho_j = -1)$$
$$q_i = P(\varphi_i = -1 | \varrho_j = -1)$$
$$\bar{q}_i = P(\varphi_i = -1 | \varrho_j = 1)$$

A maximum likelihood hányados értéke független befolyásoló tényezők esetén:

$$\frac{1}{2}\log \prod_{i=1}^{k} \left(\frac{p_i}{\bar{p}_i} \cdot \frac{\bar{q}_i}{q_i}\right)^{\varphi_i} = \frac{1}{2}\sum_{i=1}^{k} \varphi_i \log \frac{p_i \bar{q}_i}{\bar{p}_i q_i} = \sum_{i=1}^{k} \varphi_i c_i^{(0)}.$$

Ebből az egyenlőségből a

$$c_i^{(0)} = \frac{1}{2}\log \frac{p_i \bar{q}_i}{\bar{p}_i q_i}, \quad i = 1, \ldots, k$$

választás alkalmasnak mondható.

e) Az elkészült számítógépes programok rövid ismertetése.

I. PERINAT

Funkciója: Feltételes relatív gyakoriságok számítása, *u-próbával* szignifikancia szintek meghatározása, az **A** mátrix és a kezdő $\mathbf{c}_0$ előállítása, továbbá tünetkombinációk képzése (paramétertől függően).

*Bemenő paraméterek: n, m, K*1, *M*1

*K*1: a kombinálandó tünetek száma (ha nincs kombinációképzés, akkor 0).

*M*1: az **A** mátrix oszlopainak száma.

Adatok (*n* db *m*-tényezős vektor) bevitele kártyáról az (I4, I1, I2, I1, I4, 19I1, 6I2, I1) formátum szerint.

Memóriaigény: 22 *K*.

Futási idő: 3′ 44″.

II. KASHYAP

Funkciója: A *Kashyap—Ho algoritmusnak, módosított változatának* és a *lineáris regressziós módszernek* végrehajtása.

*Bemenő paraméterek: n, M*1.

Adatok: Induló $\mathbf{c}_0$ (*M*1 komponens) beolvasása 8F10.0 formátummal,

$n * M1$-es mátrix bevitele 16I4-es formátummal.

Memóriaigény: 38 *K*.

Futási idő: 13′ 24″.

III. REX programcsomag használata [9].

Az (1.6) megoldására szolgáló algoritmust a REX rendszer egyedi korlátos módosított szimplex algoritmusának alkalmazásával próbáltuk ki. Ez speciális mágnesszalagos adatelőkészítést és programírást igényelt.

Memóriaigény: 64 *K*.

Futási idő: 29′ 12″.

f) A különböző algoritmusok összehasonlítása.

Az 1. szakaszban felsorolt módszereket végrehajtó KASHYAP és REX programok által nyert eredmények összevethetők a 2. táblázat alapján:

2. TÁBLÁZAT

Módszer	Kashyap—Ho	„Regressziós"	Mód. Kash.—Ho	Lineáris progr.
súlyvektorok	.04	.04	.03	.03
	.10	.10	.04	.08
	.16	.16	.10	.15
	.24	.24	.07	.22
	.09	.09	.12	.08
	.12	.11	.02	.10
	.37	.37	.10	.35
	.21	.21	.12	.20
	.11	.11	−.09	.09
	.23	.23	.14	.21
	.05	.05	.03	.03
	.05	.05	.02	.03
	−.09	−.09	−.05	−.05
	.09	.09	.03	.07
	.07	.07	.01	.05
Diagnosztikai pontosság	70%	68%	56%	75%

A három legjobb diagnosztikai pontosságot adó algoritmus eredményei a legtöbb esetben a harmadik tizedes jegyben térnek el. A legjobb eredményt a *Kashyap—Ho* és a *lineáris programozási módszer* szolgáltatták. Ezeknek a kiterjedtebb alkalmazását érdemes lesz megpróbálni.

g) A „lényegesen befolyásoló" tényezők vizsgálata.

A 2. szakasz a) pontjában szereplő 15 tényező alaposabb elemzését végeztük el. Az egyes tényezőkhöz előzetesen hozzárendelt ismeretlen súlyokat az 1. szakaszban ismertetett különböző matematikai módszerek számítógépes realizációjával nyert programok lefuttatásával kaptuk meg. Az agyvérzést legnagyobb súllyal befolyásoló öt tényező a következő: (A súlyok összegét 100 százaléknak véve számítottuk ki az egyes súlyok százalékos arányát.)

Sorszám	Befolyásoló tényező	Súly	Százalék
1.	Toxaemia csoport (terhesség alatt kezdődött vesebetegség, praeeclampsia, ecclampsia)	0,36626	20,618
2.	Rendellenes lefolyású szülés	0,23782	13,880
3.	Placenta praevia	0,23133	13,023
4.	A szülőerők rendellenessége	0,21178	11,922
5.	Harántfekvés, medencevégű fekvés	0,16089	9,057

A súlyok szerinti rendezésben a 2. szakasz a) pontjának öt tényezője közel egyenlő (0,1) súllyal és százalékkal (5%) szerepel, a 11—15 tényezők pedig az agyvérzés kialakulásában alig jönnek számításba.

Meghatároztuk és vizsgáltuk az öt legnagyobb súllyal szereplő tényező lehetséges kombinációit $\left(\sum_{k=2}^{5}\binom{5}{k}\cdot 2^k = 232 \text{ eset}\right)$. Megállapítottuk az agyvérzés egyes kombinációkra vonatkozó feltételes relatív gyakoriságát (γ_n) és a megfelelő konfidencia intervallumot 95 százalékos megbízhatósági szint mellett. A megbízhatósági intervallumok hosszának kiszámításánál a tényleges binomiális eloszlás helyett az

$$\eta_n = \frac{n\gamma_n - np}{\sqrt{np(1-p)}}$$

közelítő normális eloszlást ($n \geqq 50$) vettük alapul, és a C_1 és C_2 intervallum határokat pedig a

$$C_1 = \frac{\gamma_n + \frac{u_\varepsilon^2}{2n} - \frac{u_\varepsilon}{\sqrt{n}}\sqrt{\gamma_n(1-\gamma_n) + \frac{u_\varepsilon^2}{4n}}}{1 + \frac{u_\varepsilon^2}{n}},$$

$$C_2 = \frac{\gamma_n + \frac{u_\varepsilon^2}{2n} + \frac{u_\varepsilon}{\sqrt{n}}\sqrt{\gamma_n(1-\gamma_n) + \frac{u_\varepsilon^2}{4n}}}{1 + \frac{u_\varepsilon^2}{n}}$$

formulákkal számítottuk ki, ahol $u = 1{,}96$, n pedig a megfelelő tünetkombináció előfordulásának gyakorisága.

3. Az eredmények rövid orvosi értékelése

Vizsgálataink szerint a koraszülöttek koponyaűri vérzésének kialakulása szempontjából legfontosabb tényező a késői *terhességi toxaemia,* illetve ennek különböző megnyilvánulási formái.

A szülés lefolyása döntő jelentőségű a koraszülöttek idegrendszeri vérzésének bekövetkezésében.

Rendellenes lefolyásúnak tartottuk a szülést, ha azt császármetszéssel, fogóval, vacuum extractorral fejezték be, ill. ha a szülészeti értékelés azt rendellenes lefolyásúnak minősítette.

A „*szülőerők rendellenessége*" összefoglaló elnevezés alatt két, egymástól szélsőségesen eltérő kórkép nyer jelentőséget: az elhúzódó szülés és a rohamos szülés.

Az *elhúzódó szülésnek* az anoxiás típusú agyvérzés keletkezésében van jelentősége;

a *rohamos szülés* inkább mechanikus károsodást okoz, uterin és extrauterin nyomáskülönbségek kiegyenlítődéséről van szó.

A *medencevégű fekvéses szülés* különösen nagy megterhelést ró a magzatra, mely különösen a magzat koponyáját veszi igénybe. A magzat traumás koponyaűri károsodását eredményezheti a falx cerebri, a tentorium cerebelli szakadásával, következményes subduralis vérzéssel.

A *harántfekvés* jelentősége a traumás típusú idegrendszeri vérzések keletkezésében van.

Jelentős tényezőnek bizonyult a koraszülöttek koponyaűri vérzésének szempontjából, ha a szülést a magzat *expressiojával, extractiojával* fejezték be.

Az *anya pyelitise, pyelonephritise* a magzat anoxiás károsodásának létrejöttében jelentős tényező.

Már az esetek kiválasztásánál határok közé szorítottuk a születési súlyt: az 1000 g alatti súlyúakat eleve kizártuk a vizsgálatból. Így is realizálódott az az irodalmi megállapítás, mely szerint az alacsonyabb súlyú újszülöttekben gyakoribb koponya ürivérzés, mint a nagyobb súlyúakban.

A *császármetszés* je]entőségét illetően a koraszülöttek koponyaűri vérzéseinek keletkezésében azon szerzők véleményével értünk egyet, akik szerint azon betegségek, melyek a császármetszést indikálták — gyakran már műtét előtt — koponyaűri vérzéshez vezethetnek.

Az *anya életkora* csak mint idős primipara jön szóba a koponyaűri vérzések keletkezése szempontjából.

A *művi vetélések* szerepe a koraszülött-halálozásban jól ismert. Az arteficialis abortusok hatása a koponyaűri vérzések kialakulására csak többszörös áttételek útján érvényesülhetne. A szülészek előtt jól ismert azonban, hogy az anamnaesis felvételekor — az adatlapoknak ezt a részét bemondás alapján töltik ki — a gravidák szívesen hallgatják el korábbi művi vetéléseiket. Mindezen tényezők hatásának tudható be, hogy a kórelőzményben szereplő művi terhességmegszakítások és koponyaűri vérzés között nem találunk szignifikáns korrelációt. A művi terhességmegszakítások leggyakoribb szövődménye a méhnyak elégtelenség, amely viszont objektív tünet,

3. TÁBLÁZAT

Az agyvérzés szempontjából legfontosabbnak bizonyult öt tényező néhány jellemző kombinációja

A kombináció azonosítási száma	Medencevégű, harántfekvés	Rendellenes lefolyású szülés	Toxaemia csoport	Szülőerők rendellenessége	Placenta praevia	Kombináció gyakorisága	Koponyaűri vérzés relatív gyakorisága	Konfidencia
3	—	—	—	—	—	123	0,34	0,16
119	—	—	—	Ø	Ø	164	0,35	0,14
157	Ø	—	—	—	Ø	203	0,37	0,13
195	—	—	Ø	Ø	Ø	201	0,40	0,13
227	Ø	—	+	Ø	Ø	51	0,52	0,26
4	—	+	+	—	—	71	0,70	0,20
194	+	+	Ø	Ø	Ø	82	0,60	0,20
95	—	Ø	+	—	—	103	0,66	0,17
111	Ø	+	+	—	—	73	0,71	0,20
46	—	+	—	+	Ø	76	0,69	0,20
62	—	+	+	Ø	—	75	0,72	0,19
81	+	Ø	+	+	+	103	0,66	0,17

Jelmagyarázat: + tényező jelen van
— tényező nincs jelen
Ø tényezőt nem vettük tekintetbe

melyet az orvos állapít meg, lényeges tényezőnek bizonyult az újszülöttek koponyaűri vérzésének létrejöttében, mely mutatja, hogy a művi vetélések hatása valóban csak áttételesen érvényesül.

Az öt kiszemelt, súlyozott tényező kombinációit (3. táblázat) vizsgálva megállapítottuk, hogy a feldolgozás során igen nagy súlyt kapott toxaemia csaknem bármely kombinációban érezteti jelentős hatását. A táblázatban feltüntetett kombinációkból láthatjuk, hogy ha a kiemelt öt tényező egyike sincs jelen, nagy esetszám mellett az agyvérzés feltételes relatív gyakorisága még a 0,5-öt sem éri el és ez a megállapítás szűk konfidencia határok között érvényes, tehát megbízható.

A toxaemia rendes lefolyású szülés mellett is szerepet játszik a koponyaűri vérzés keletkezésében, de ha rendellenes lefolyású szüléssel társul, az agyvérzés feltételes relatív gyakorisága nagyobb s ez a megállapításunk is nagyobb megbízhatóságú.

Rendes lefolyású szülés — ha más tényező nem szerepel — ritkán vezet agyvérzéshez. Medencevégű fekvés már az esetek felében agyvérzést „okozhat". Ha azonban rendellenes lefolyású szüléssel és medencevégű fekvéssel járó eseteinket vizsgáljuk, a koponyaűri vérzés gyakorisága még tovább emelkedik.

A szülési tényezőknek a perinatalis mortalitásban játszott szerepét többen vizsgálták, többnyire azonban csak egy-egy tényező szerepét vették tekintetbe egyszerre, azonkívül meglehetősen kisszámú mintán végezték a vizsgálatokat. Feldolgozásunk jelentőségét abban látjuk, hogy igen nagy esetszám mellett, de mégis szelektált anyagot vizsgáltunk. Egyszerre több tényező szerepét vettük tekintetbe, s az egyes tényezőket súlyoztuk.

Köszönetnyilvánítás. E helyen mondunk köszönetet a Semmelweis Orvostudományi Egyetem Számítástechnikai Csoportja munkatársainak, Mayer Jánosné tudományos munkatársnak és Keszthelyi Éva tudományos segédmunkatársnak, akik a különböző matematikai módszerek számítógépes realizálásában segítségünkre voltak, továbbá dr. Szigeti Ferencnének és Cséka Évának az adatelőkészítésben végzett gondos munkájukért.

IRODALOM

[1] DANTZIG, G. B., *Linear Programming and Extensions* (Princeton University Pess, Princeton, New Yersey, 1963).

[2] FRITZ, J., "On a pattern classificatin algorithm of R. L. Kashyap", *Problems of Control and Information Theory* **2** (1973) 81—92.

[3] HO, Y. C. and KASHYAP, R. L., "A class of iterative procedures for linear inequalities", *J. SIAM Control* **4** (1966) 112—115.

[4] HO, Y. C. and KASHYAP, R. L., "A class of iterative procedures for linear inequalities", *JEEE EC* **14** (1965) 683—688.

[5] HOEL, P. G., *Introduction to Mathematical Statistics* (John Wiley and Sons, Inc. New York, 1971).

[6] MEISEL, W. S., *Computer-oriented Approach to Patter Recognition* (Academic Press, New York, 1972).

[7] MENDEL, J. M. and FU, K. S., *Adaptive, Learning and Pattern Recognition Systems* (Academic Press, New York, 1970).

[8] PRÉKOPA, A., *Lineáris programozás* (A Bolyai János Matematikai Társulat kiadványa, Budapest, 1973).

[9] ORCHARD-HAYS, W., *Advenced Linear Programming and Computing Techniques* (McGraw-Hill Book Co., 1968).
[10] VINCZE, I., *Matematikai statisztika ipari alkalmazásokkal* (Műszaki Könyvkiadó, Budapest, 1968).

(Beérkezett: 1976. május 5.)
(Újra beérkezett: 1976. szeptember 16.)

STRAJBER BENEDEK
SOTE SZÁMÍTÁSTECHNIKAI CSOPORT
1089 BUDAPEST, KULICH GYULA TÉR 5.
FRITZ JÓZSEF
MTA MATEMATIKAI KUTATÓ INTÉZET
1053 BUDAPEST, REÁLTANODA U. 13/15.
SEBŐK JÁNOS
ORSZÁGOS KÓRBONCTANI ÉS KÓRSZÖVETTANI INTÉZET
1389 BUDAPEST, SZABOLCS U. 35.
PAKSY ANDRÁS
SOTE BIOMETRIAI CSOPORT
1083 BUDAPEST, KORÁNYI S. U. 2/A.
KISZEL JÁNOS
SOTE I. SZ. NŐI KLINIKA
1088 BUDAPEST, BAROSS U. 27.

THE APPLICATION OF MATHEMATICAL METHODS FOR THE DISCRIMINATION OF TWO CLASSES OF DISEASES

B. SRAJBER, J. SEBŐK, J. FRITZ, A. PAKSY, J. KISZEL

We describe linear discriminating methods for separating two classes of diseases. The efficiencies of the four different processes were compared by computing realisations. We applied the methods for the investigation of the causes of intracranial haemorrhage in connection with premature infants (1000 cases, 57 factors one by one).

Alkalmazott Matematikai Lapok **2** *(1976) 233—235*

EGY OSZTÁLYOZÁSI FELADAT MEGOLDÁSA

HEPPES ALADÁR, MÁLYUSZ KÁROLY, STAHL JÁNOS
Budapest

A postahivatalok munkájának az automatizálása a következő problémát vetette fel. Adott a levelek n osztálya és ismert az i-ik osztályban levő levelek a_i száma, $i=1, \ldots, n$. Rendelkezésre áll osztályozógépek egy halmaza, melyek közül a j-ik gép a leveleket egy művelettel m_j csoportba tudja szétosztani. Mindegyik csoport a levélosztályok egy részhalmazából tevődik össze. Az m_j számok minden osztályozógépre rendelkezésre állnak. A dolgozatban leírunk egy algoritmust, mely meghatározza azt az osztályozási sémát, amely a végső osztályokat a lehető legkevesebb munkával állítja elő. Ez utóbbit az osztályozáshoz szükséges teljes műveletszámmal mérjük.

A postai küldemények számának növekedése egyre nagyobb problémát jelent a feldolgozó hivatalok számára. A megoldást természetesen a gépesítés, automatizálás különféle formáiban keresik. A levelek címszerinti válogatását könnyítő osztályozó automaták gazdaságos kihasználásával kapcsolatos a jelen dolgozatban tárgyalt matematikai probléma.

Osztályozó gépen olyan berendezést értünk, amely a rajta áthaladó levelet a levélen levő jelzés, pl. irányítószám alapján m számú osztályba sorolja a gépen beállítható képzési szabály szerint. (A gép kapacitásán az m számot értjük). Ha a leveleket m-nél több felé akarjuk válogatni — ami általában a helyzet —, akkor az első válogatásnál keletkező egyes osztályokat (esetleg valamennyit) ismételt osztályozással — persze más képzési szabály mellett — tovább kell válogatni.

Általában ismertnek tételezhető fel azoknak a leveleknek a száma (vagy legalábbis a várható száma), amelyek az egyes végső, tovább már nem osztandó osztályokba tartoznak. Feladatunk ennek és az m kapacitásnak ismeretében olyan küzbülső osztályképzési szabály, osztályozási séma meghatározása, amely mellett a teljes géphasználat minimális.

A problémát annyival általánosabban fogjuk tárgyalni, hogy megengedjük az egyes osztályozási műveletekben eltérő osztályozó kapacitású berendezések használatát.

Legyenek $a_1 \geqq a_2 \geqq \ldots \geqq a_n > 0$, ahol a_i az i-ik tulajdonságú elemek száma, $m_1 \geqq m_2 \geqq \ldots \geqq 2$ pedig a választékul rendelkezésre álló osztályozó egységek kapacitásai. A j-edik egység tehát az elemeket tulajdonságaik alapján és a megadott osztályképzési szabály szerint m_j számú osztályba képes sorolni. Minthogy kisebb kapacitású osztályozó egység mindig pótolható még fel nem használt nagyobbal, feltehetjük, hogy az osztályozás során az első s számú egység kerül felhasználásra.

Egy osztályozási sémát természetes módon feleltethetünk meg egy fa alakú irányított gráfnak, amelynek élei osztályoknak (végpontba futó élei végső osztályoknak), elágazási pontjai pedig egy-egy osztályozó egységnek felelnek meg. Ha egy egység kapacitása m_i, akkor a neki megfelelő pontból legfeljebb m_i számú él fut ki.

Az elágazási pontba befutó élhez tartozó osztály mint halmaz a pontból kifutó élekhez tartozó idegen halmazok egyesítésével megegyezik. A kezdőponttól a gráf más pontjaihoz vezető utak éleinek számát az illető pont szintjének nevezzük. Az egyes pontokba befutó él által képviselt osztály elemei előzőleg a pont szintjének megfelelő számú osztályozáson estek át. A minimalizálandó teljes műveletszám a végpontok szintszámainak és a befutó élhez tartozó elemszámoknak a szorzataiból képezett összeg.

1. *Megjegyzés.* Optimális osztályozási sémában a legmagasabb szint kivételével nem fordulhat elő olyan elágazási pont, amelyből a kapacitásánál kevesebb számú él fut ki. (Ellenkező esetben ugyanis csökkenthető a teljes műveletszám azáltal, hogy egy nagyobb szintű elágazási pontból kifutó osztályt az alacsonyabb szinten levő kihasználatlan kapacitású elágazási ponthoz sorolunk. Az átsorolás természetesen értelemszerűen megváltoztatja a kezdőpontból az érintett csomópontokhoz vezető utak mentén elhelyezkedő éleknek megfelelő osztályokat is.)

2. *Megjegyzés.* Ha egy optimális sémában két *elágazási pont* szintszáma különböző, az alacsonyabb szintűből legalább annyi élnek kell kifutnia, mint a magasabb szintűből. (Ha a nagyobb szintűből k-val több él futna ki — $k>0$ —, akkor k számú kifutó él a belőlük elérhető részgráfokkal együtt átsorolható volna a kisebb szintszámú elágazási ponthoz a pontok kapacitásának egyidejű felcserélése mellett. Ez az átrendezés csökkentené a teljes műveletszámot.)

3. *Megjegyzés.* Ha egy optimális sémában két *végpont* különböző szinten helyezkedik el, akkor az alacsonyabb szintűhöz tartozó osztály legalább annyi elemű, mint a magasabb szintűhöz tartozó. (Ellenkező esetben a két osztály szerepcseréje csökkentené a teljes műveletszámot.)

4. *Megjegyzés.* Ha az azonos szintű végpontokhoz tartozó osztályhozzárendelést átrendezzük, a teljes műveletszám nem változik.

A továbbiakban a 4. megjegyzés alapján — az alternatív optimumok egy részének kizárásával — feltételezzük, hogy az 1. megjegyzésben említett legmagasabb szintű elágazási pontok között legfeljebb egy, a legnagyobb indexű kapacitása nincs teljesen kihasználva.

5. *Megjegyzés.* Egy optimális megoldásban a legnagyobb indexü — esetleg kihasználatlan kapacitásu — elágazási pontból kifutó élek számát k-val jelölve az alábbi összefüggés áll fenn:

$$k+\sum_{i=1}^{s-1}(m_i-1)=n, \quad 2\leqq k\leqq m_s$$

valamely s indexre, ami s és k értékét egyértelműen meghatározza.

(A képlet igazolására számoljuk össze az n végpontot a legmagasabb indexü elágazási pont törlésével kezdve. Ha egy ilyen pontot a belőle kiinduló (végpontokba futó) élekkel együtt töröljük a gráfból, az új gráfban (m-1)-gyel kevesebb végpont lesz, ha a törölt csúcsból m él futott ki. Az eljárást folytatva végül 1 pont marad, ez pedig az összefüggés helyességét igazolja. Az összefüggésből nemcsak k értéke, hanem az elágazások s száma is meghatározható.)

6. *Megjegyzés.* Egy optimális megoldásban tekintsük a legnagyobb indexű elágazási pontot. Egyesítsük az ehhez az elágazási ponthoz tartozó osztályokat egyet-

len osztályba és az így kapott osztályba tartozó elemek száma legyen a megfelelő elemszámok összege. Töröljük a megoldáshoz tartozó fa (szóban forgó, azaz) legnagyobb indexű elágazási pontjából kifutó éleket és az új végpontnak feleljen meg a szóban forgó összeg. Az így kapott fa a redukált feladat optimális megoldását reprezentálja.

(Ugyanis a redukált feladat bármely megoldására a redukciós lépés fordítottját alkalmazva az eredeti feladat egy megoldásához jutunk. A műveletszám változása minden esetben ugyanannyi, nevezetesen az egyesítéssel adódott osztályhoz tartozó elemszám. Következésképpen a redukált feladatnak nem lehet annál jobb megoldása, mint ami az eredeti feladat optimális megoldásának redukciójával adódik.)

Optimális osztályozási sémát határozhatunk meg az alábbi algoritmussal:

1. *lépés:* k és s értékének meghatározása az 5. megjegyzés alapján, legyen továbbá $m_s=k$.

2. *(redukciós) lépés:* rendeljük az s indexű elágazási ponthoz az eddig még be nem sorolt végső osztályok közül az m_s legkisebb elemszámút. Egyesítsük ezeket egy osztállyá és feleltessük meg a befutó élnek. Tekintsük most az így létrehozott osztályt végső osztálynak, hagyjuk el az osztályok listájáról ennek összetevőit és csökkentsük s értékét 1-gyel. Ha $s=0$, az eljárás befejeződött, ha nem, ismételjük a 2. lépést.

Az algoritmus verifikálása a korábbi megjegyzésekből teljes indukcióval adódik.

(Beérkezett: 1976. június 14.)

HEPPES ALADÁR, MÁLYUSZ KÁROLY ÉS STAHL JÁNOS
SZÁMÍTÓGÉPALKALMAZÁSI KUTATÓ INTÉZET
1015 BUDAPEST I., CSALOGÁNY U. 30—32.

ON THE SOLUTION OF A SORTING PROBLEM

A. HEPPES, K. MÁLYUSZ and J. STAHL

The automatization of the work in postal offices posed the following problem.

There are n classes of letters and the number of the letters in the i-th class a_i is given. There is a pool of sorting machines and the j-th one is able to sort the letters into m_j groups in one operation, where each group contains a subset of classes. The numbers m_j are also known.

An algorithm is given to determine the sorting rules and hierarchy that provide the sorting with the least sorting work. This work is measured by the total number of sortings for all letters.

Alkalmazott Matematikai Lapok 2 (1976) 237—257

FOLYAMATOK KVALITATÍV VIZSGÁLATÁRÓL

FARKAS MIKLÓS

„Semmiféle emberi kutatás nem tekintheti magát igazi tudománynak, ha matematikailag nem nyert igazolást."

Leonardo da Vinci

1. Bevezetés

Leonardo mottóul választott mondását abszolút igazságnak tekintem. Ugyanakkor belátom, hogy ez a hosszú távon feltétlenül érvényes igazság rövid távon és viszonylagosan igazságtalan lehet.

A biológia tudományának kifejlődése során több évszázados adatgyűjtő, rendszerező és kísérletező tevékenység előzte meg DARWIN, MENDEL, PAVLOV, WATSON és CRICK fellépését. Az évszázadok alatt felhalmazódott adattömeg tette lehetővé a felsorolt nevekhez fűződő, általános rendező elveknek és törvényszerűségeknek felismerését. És csak most tartunk ott, hogy elérhető közelségbe került olyan matematikai modellek megalkotása, melyek az objektív törvények lényegét képesek visszatükrözni. Kezd kirajzolódni az „elméleti biológia", vagyis az a biológia, mely a tapasztalati adatok alapján képes a lényeget bizonyos számú „axiómában" megragadni, az „axiómák" rendszerét matematikai alakba önteni, az így keletkezett matematikai modellből deduktív úton kvantitatív és kvalitatív következtetésket levonni és ellenőrizni a kapott eredmények egyezését a kísérleti adatokkal. A biológia tehát „igazi tudománnyá" válik. Méltánytalanság lenne azonban azt állítani, hogy a XVII. és XVIII. század nagy botanikusai és zoológusai nem „igazi tudománnyal" foglalkoztak, bár — részben a matematika fejletlensége miatt — kevéssé alkalmaztak matematikai módszereket.

Évezredek történetírása előzte meg MARX és ENGELS fellépését. Az így felhalmozódott adattömeg és tapasztalat tette lehetővé azt, hogy felismerjék az emberi társadalom mélyén ható, alapvető, objektív törvényeket, el tudják választani a véletlent a szükségszerűtől, a meghatározót a meghatározott-tól. MARX és ENGELS működése nyomán a történettudomány leíró tudományból elméleti tudománnyá vált, és — úgy tűnik — nem utópia a történelmi folyamatok lényegét megragadó matematikai modellek megalkotásának lehetősége.

„Igazi tudomány" lehet tehát valamilyen tárgyra, jelenségre irányuló kutatás akkor is, ha kialakulásának kezdetén és esetleg még hosszú időn át képtelen a matematika alkalmazására. Ami azonban „igazi tudomány", az előbb—utóbb „matematizálhatóvá" válik.

Ez a megállapítás merésznek tűnhet és néhány évtizeddel ezelőtt nehezen lett volna védhető. Ma azonban amellett, hogy igazságát alátámasztja a mechanika, a csillagászat, a fizika, a kémia, a biológia, a szabályozáselmélet, a közgazdaságtudomány, a demográfia, a nyelvtudomány, a hadtudomány, a pszichológia fejlődése, figyelembe kell venni magának a matematikának az utóbbi néhány évtizedben

és különösen az elmúlt néhány évben elért eredményeit. Ez a fejlődés oda vezetett, hogy ma már a *matematika nem csupán a jelenségek kvantitatív, hanem kvalitatív leírására is alkalmas.*

Félreértések elkerülése végett meg kell jegyezni, hogy matematikai apparátus felhasználása önmagában nem avat valamely szellemi tevékenységet tudománnyá. A számmisztika, mely különböző korokban időről időre felvirágzik, nyilván nem tudomány, pedig a matematikán „alapul". A néhány éve kialakulóban levő futurológia kezdettől fogva rendkívül erős matematikai apparátust használ. Azonban az, hogy tudomány-e, nem ezen múlik, hanem azon, hogy eredményei az objektív valóság valamely oldalát megközelítőleg hűen tükrözik-e, illetve azon, hogy a jövő egyáltalán az objektív valóság részének tekinthető-e. Jelen tanulmány szerzője nem érzi magát felkészültnek ennek eldöntésére.

A következőkben megkísérlem vázolni azt a részben még kialakulóban levő matematikai apparátust és módszert, mely meggyőződésem szerint éppen napjainkban új helyzetet teremt az objektív valóságról, a világról alkotott tudományos képünk fejlődésében.

Elsősorban az időben lejátszódó folyamatokat tartom szem előtt. Az idő szerepeltetése egyetemes paraméterként némileg megszorítja az általánosságot. A földi légkör nyomásának változása (meghatározott időpontban, egy földrajzi koordinátákkal adott helyen) a tengerszint feletti magasság függvényében például ekkor nem fér be a vizsgált jelenségek körébe. Hasonlóan kimarad az érvényességi körből annak vizsgálata, hogyan függ valamely ország bűnözési statisztikája az egy főre jutó alkoholfogyasztástól. Az ilyen típusú vizsgálatokra azonban az elmélet könnyen és kézenfekvő módon kiterjeszthető. Ugyanakkor az, hogy az időt tekintjük paraméterünknek, a szóhasználat egyszerűsödése mellett több előnnyel jár.

Először, az amúgy is meglehetősen absztrakt módszert valamelyest kézzelfoghatóbbá, konkrétabbá teszi és az általánosság leszűkítése ellenére is a legizgalmasabb alkalmazási területek és jelenségek tömegeit felöleli.

Másodszor, a fizikai értelemben vett időt matematikailag egy kezdeti időpont és egy időegység megválasztásával a valós számok R halmazával írjuk le, az R halmazt viszont a számegyenessel ábrázoljuk. A számegyenes, mint matematikai absztrakció „két irányban járható be". A fizikai értelemben vett idő, azonban csak egy irányban változik, telik, múlik. Ha a négydimenziós tér-időben t-vel jelöljük az időt és x-szel a helyet és a (t_0, x^0) állapot megváltozik (t_1, x^1)-re, ahol $t_1 > t_0$, akkor az x^0 helyhez tartozó állapot még ismét létrejöhet, de ismét a t_0 időpillanathoz tartozó állapot már nem. Ennek a trivialitásnak fontos matematikai következménye az, hogy ha valamely determinisztikusnak tekinthető jelenséget meghatározó paraméterek között az idő is szerepel, ez utóbbira mint független változóra a jelenség mindig „felfűzhető", a többi meghatározó paraméter és magának a jelenségnek a lefolyása az időtől függő *függvénnyel* leírható. (Függvény-fogalmunk lényege az, hogy a független változó egy szóba jövő értékéhez pontosan egy függvényérték tartozzék). A korábban, más összefüggésben említett példákból kitűnik az időnek, mint egyetemes paraméternek kitüntetett szerepe. Ha megmérjük a légnyomást egy időpontban egy helyen, majd egy későbbi időpontban egy másik helyen, ezután még visszatérhetünk az első helyre, de nem tudunk visszajutni az első időponthoz. Határozzuk meg egy adott évben az egy főre eső alkoholfogyasztást és az abban az évben elkövetett bűntények számát, majd tegyük meg ugyanezt egy későbbi év-

ben is (és akkor legyen az alkoholfogyasztás értéke az előzőtől különböző). Ezután előfordulhat az, hogy egy későbbi vizsgálatnál ugyanaz az alkoholfogyasztási érték adódik, mint először (de más bűnözési statisztika), az első vizsgálat évét azonban többé nem reprodukálhatjuk. A bűnözési statisztika tehát nem adható meg az alkoholfogyasztás függvényeként (e két dolog között nyilván sztochasztikus a kapcsolat), megadható viszont az idő függvényében.

Szeretném elérni azt, hogy ezt a dolgozatot nem matematikus, tudományos kutatók is megértsék. Ezért igyekszem a matematikai formalizmust minimálisra csökkenteni, a matematikailag fontos, de a lényeget nem érintő feltételek felsorolását elhagyom és pontos matematikai megfogalmazások helyett, ahol csak lehet, megelégszem a köznapi nyelvből vett szavak intuitív értelemben vett felhasználásával.

2. Dinamikai rendszerek és attraktoraik

Lássunk néhány példát olyan jelenségre, folyamatra, amelynek matematikai leírására, modellezésére törekszünk. Tömegpontnak tekinthető test mozgása a Föld nehézségi erőterében. Kémiai reakció lezajlása több reagens között, amennyiben a jelenséget nem pillanat alatt lejátszódónak tekintjük, hanem az idő függvényében kívánjuk követni. Valamely ország gazdasági életének alakulása. Az embrió fejlődése, vagy általánosabban az ontogenezis. Valamely biológiai populáció genotípus-összetételének fejlődése. A filogenezis. Egyazon ökológiai környezetben élő fajok számarányának alakulása. Egy személy lelkiállapotának változása az időben. A tanulás folyamata. Járványok terjedése.

Ha egy folyamat matematikai modelljét kívánjuk megalkotni, az első feladat annak eldöntése, hogy a folyamat determinisztikus-e. Determinisztikusnak nevezünk egy folyamatot, ha pillanatnyi állapota egyértelműen meghatározza állapotát bármelyik jövőbeli és múltbeli időpillanatban. Vannak olyan jelenségek, elsősorban az elemi részecskék fizikájában, amelyek jelenlegi tudásunk szerint, elvileg nem lehetnek determinisztikusak. Ezeket az egyszerűség kedvéért, egyelőre kirekesztjük a tárgyalásból. A makrovilág jelenségeit azonban determinisztikusnak tekinthetnénk, ha elég sok adattal tudnánk jellemezni az állapotukat. Tipikus valószínűségszámítási feladat a játékkocka feldobása után bekövetkező események analízise. Pedig ha vízszintes lap fölött, légüres térben dobjuk fel a kockát és az elengedés pillanatában pontosan megadjuk a kockának (mint merev testnek) a kezdeti állapotát (helyzet, a súlypont sebessége, szögsebesség), előre megmondhatnánk, hányast dobunk. Arra, hogy a kockadobás eredményét miért tekintjük mégis véletlen eseménynek, később visszatérünk.

Az érdekes és fontos jelenségek jelentős része a *lényeget tekintve determinisztikus, a mellékes körülményeket tekintve sztochasztikus.* Azt, hogy egy kiszemelt vízmolekula milyen pályán kerül vissza a tengerbe, számtalan figyelembe nem vehető körülménytől függ, de az, hogy visszakerül, „biztosra vehető", hacsak valami rendkívüli nem történik vele (befagy az Antarktiszon, kilövik a Földről egy Mars-szondán, megszűnik létezni disszociáció útján stb.). Egészséges emberi zigotából, egészséges anyai szervezetben és normális külső körülmények között emberi újszülött, majd békés emberi társadalomban, civilizált körülmények között a leglényegesebb biológiai tulajdonságokban meghatározott fenotípusú felnőtt ember lesz. Ezt a lényeges vonatkozásokban determinált felnőtt fenotípust egypetéjű ikrek különböző módon, más-

más „pályán" érhetik el például annak következtében, hogy különböző körülmények között nevelkednek. Természetesen bizonyos mértékig szubjektív dolog annak eldöntése, hogy mi lényeges és mi nem az. Továbbá előfordulhatnak balesetek, „katasztrófák", melyek során az egypetéjű ikrek egyike krónikus betegségre tesz szert, vagy meghal.

A determinisztikus, vagy a lényeget tekintve determinisztikus folyamatok matematikai leírásának alapvető eszköze a „dinamikai rendszer" fogalma [8]. Tételezzük fel, hogy a vizsgált jelenség állapota bármelyik időpontban n számú adattal az $x_1, x_2, \ldots, x_n$ valós számokkal jellemezhető az $x=(x_1, \ldots, x_n)$ vektorok R^n halmazát *állapottérnek,* vagy *fázistérnek* nevezzük. Az, hogy a rendszer determinisztikus, a következőket jelenti. Ha a rendszer állapota például a $t=0$ időpontban $x^0 \in R^n$, akkor bármely más t időpontban ismerjük a rendszer $x \in R^n$ állapotát. Más szóval minden t időponthoz (valós számhoz) tartozik egy leképezés, mely az R^n állapotteret önmagába képezi le úgy, hogy az R^n tér tetszőleges x^0 pontjához hozzárendeli az R^n tér azon x pontját, mely a rendszer állapotát jellemzi a t időpontban, feltéve, hogy a rendszer a $t=0$ időpontban az x^0 állapotban volt. Valójában tehát egy $n+1$ változós φ függvény van adva, mely minden t időponthoz és x^0 állapothoz hozzárendel pontosan egy $x=\varphi(t, x^0)$ állapotot. A φ függvény bizonyos alapvető tulajdonságokkal rendelkezik. Ezek közül az egyik legfontosabb az, hogy fennáll a

$$\varphi(t+s, x^0) \equiv \varphi\big(s, \varphi(t, x^0)\big) \tag{1}$$

azonosság. Ugyanis a jobboldalon a rendszer állapota áll az s időpontban, ha a kezdőpillanatban az állapota $\varphi(t, x^0)$ volt; a baloldalon viszont a rendszer állapota áll a $t+s$ időpontban, ha a kezdőpillanatban az állapota x^0 volt. Ha azonban a kezdőpillanatban az állapot x^0, akkor a t időpontban az állapot $\varphi(t, x^0)$, és további s idő múlva a rendszer ugyanabba az állapotba kerül, mint amibe s idő alatt az a rendszer került, mely a kezdőpillanatban volt a $\varphi(t, x^0)$ állapotban. Ez az azonosság láthatólag azt is kifejezi, hogy a rendszer x^0 állapota egyértelműen meghatározza a rendszer állapotát tetszőleges s idővel később, *függetlenül attól, hogy mikor következett be az x^0 állapot.* Az R^n állapottérből és az előbbiekben nagyjából körvonalazott φ függvényből álló (R^n, φ) párt nevezzük *dinamikai rendszernek.*

Ha az x^0 pontot lerögzítjük, akkor a φ függvényből egy egyváltozós függvényt kapunk, mely minden t időponthoz hozzárendeli a $\varphi(t, x^0) \in R^n$ pontot. Ezt az egyváltozós függvényt φ_{x^0}-lal jelöljük $\big(\varphi_{x^0}(t)=\varphi(t, x^0)\big)$ és az x^0 pont *mozgásának* nevezzük. A φ_{x^0} függvény az R halmazt R^n-be képezi; R képhalmazát R^n-ben, vagyis a $\varphi(t, x^0)$ pontok halmazát (rögzített x^0 mellett t végigfut a valós számokon) az x^0 pont *pályájának,* vagy *trajektóriájának* nevezzük. Az x^0 pont pályája egyben minden e pályán rajta levő pont pályája is.

Ha az $a \in R^n$ pont olyan, hogy minden t-re $\varphi(t, a) \equiv a$, akkor a dinamikai rendszer *egyensúlyi helyzetének* nevezzük. Ha a rendszer valamely időpillanatban az a egyensúlyi helyzetben van, akkor ott is marad. Egyensúlyi helyzet pályája egyetlen pontból áll, ez a ponttrajektória.

Ha a φ függvényben a t változó értékét valamely T értékben lerögzítjük, akkor egy függvényt kapunk, mely az R^n teret önmagába képezi le. Ezt a függvényt φ_T-vel jelöljük: $\varphi_T(x^0)=\varphi(T, x^0)$, vagyis e függvény az x^0 ponthoz x^0 pályájának azt a pontját rendeli hozzá, melyben a rendszer a $t=T$ időpontban van, ha a $t=0$ időpontban x^0-ban volt.

Legyen $T \neq 0$ és tételezzük fel, hogy a φ_T-leképezés x^0-at x^0-ba viszi át, vagyis $\varphi_T(x^0) = \varphi(T, x^0) = x^0$. Ekkor az x^0 pontot a φ_T leképezés *fixpontjának* nevezzük. Ha valamely $T \neq 0$-ra x^0 fixpont, akkor x^0 mozgása periodikus mozgás T periódussal és x^0 pályája zárt görbe. Az (1) azonosság szerint ugyanis $\varphi(t+T, x^0) \equiv \equiv \varphi(t, \varphi(T, x^0)) \equiv \varphi(t, x^0)$. Egyensúlyi helyzet minden $T \neq 0$ számra a φ_T leképezés fixpontja, mozgása periodikus és periódusa minden valós szám, pályáját a pont-trajektóriát is zárt görbének tekintjük.

Az R^n tér H részhalmazát a dinamikai rendszer *invariáns halmazának* nevezzük, ha minden T-re a φ_T leképezés a H halmazt önmagába képezi le, vagyis ha minden pontjával együtt a pont teljes pályáját is tartalmazza. A legegyszerűbb invariáns halmazok maguk a pályák. Különösen fontos szerepet töltenek be a korlátos és zárt, röviden *kompakt* invariáns halmazok. Az egyensúlyi helyzetek és a periodikus mozgásoknak megfelelő zárt pályák ilyenek.

A dinamikai rendszer kompakt invariáns halmazát *attraktornak* nevezzük, ha stabilis, vagyis ha van olyan környezete, melynek bármely pontjából induló pálya pontjainak távolsága e halmaztól zérushoz tart $t \to \infty$ esetén. Azt a környezetet, ahonnan az attraktor „magához vonza a pályákat", az attraktor *medencéjének*, vagy *vonzási (attraktivitási) tartományának* nevezzük. Tehát az, hogy A attraktor és $M(A)$ az ő medencéje, a következőket jelenti. Az A halmaz korlátos, zárt és minden $x^0 \in A$ ponttal együtt minden $t \in R$-re $\varphi(t, x^0) \in A$; továbbá az $M(A)$ halmaz nyílt, $A \subset M(A)$, és minden $x^0 \in M(A)$-ra a $\varphi(t, x^0)$ pont távolsága A-tól zérushoz tart, ha t végtelenhez tart. A dinamikai rendszer attraktorainak medencéi az R^n állapottér nyílt részhalmazai, melyeket egymástól a tipikus esetekben zárt, nullmértékű halmazok, hiperfelületek, az ún. *szeparatrixok* határolnak el.

A dinamikai rendszer legfontosabb kvalitatív jellemzői: az attraktorok száma, jellege, elhelyezkedése és medencéik nagysága. Ezek határozzák meg a rendszer jövőjét, bizonyos átmeneti idő után, végső fokon a rendszer állapotát.

A dinamikai rendszereket a gyakorlatban előforduló esetek nagy részében *autonóm differenciálegyenlet-rendszerek* generálják. Az autonóm differenciálegyenlet-rendszer általános alakja

$$\dot{x} = f(x), \tag{2}$$

ahol a pont a t idő szerinti deriválást jelöli és f egy az R^n állapottéren értelmezett vektormező. A (2) egyenlet jobb oldalán álló függvény értéke az x pontban, az $f(x)$ vektor megadja a megoldásfüggvény időbeli változásának sebességvektorát akkor, amikor a megoldás éppen az x értéket veszi fel. A (2) rendszert azért nevezzük autonómnak, mert az állapotváltozás sebességének nagysága és iránya csak magától az állapottól függ és nem függ az időponttól, az időben esetleg változó külső körülményektől. A (2) differenciálegyenlet rendszer az (R^n, φ) dinamikai rendszert generálja, ha minden x^0-ra a φ_{x^0} függvény (vagyis x^0 mozgása) (2) megoldása, vagyis

$$\dot{\varphi}(t, x^0) \equiv f(\varphi(t, x^0)).$$

Az f függvénynek természetesen bizonyos simasági és egyéb feltételeket is ki kell elégítenie ahhoz, hogy (2) egy dinamikai rendszert generáljon, e feltételek felsorolásától azonban itt eltekinthetünk.

Az $x \equiv a$ állandófüggvény (2)-nek pontosan akkor megoldása, ha a (2) által generált dinamikai rendszernek egyensúlyi helyzete. Így annak feltétele, hogy az állapot-

tér a pontja egyensúlyi helyzet legyen, az $f(a)=0$ egyenlet teljesülése. A (2) által generált dinamikai rendszer egyensúlyi helyzeteinek meghatározása tehát viszonylag egyszerű feladat; az $f(x)=0$ egyenlet összes megoldását kell csupán előállítani. A nemállandó periodikus megoldások meghatározása, sőt már annak eldöntése, hogy egyáltalán van-e zárt pálya (az egyensúlyi helyzetektől eltekintve), sokkal bonyolultabb feladat.

Ha például (2) állandó együtthatós, homogén lineáris differenciálegyenletrendszer: $\dot{x}=Bx$, ahol B n-edrendű négyzetes mátrix és a B mátrix reguláris, akkor egyetlen egyensúlyi helyzet van: $x\equiv 0$. Ha a B mátrix stabilis, vagyis összes sajátértékének valós része negatív, akkor a 0 egyensúlyi helyzet a rendszer egyetlen attraktora, melynek medencéje az egész R^n tér.

Ha egy időben lejátszódó folyamat determinisztikus, véges sok, n számú adattal jellemezhető, és sikerül a folyamat lezajlását meghatározó objektív törvényszerűségeket (2) típusú differenciálegyenlet rendszerrel megadni, akkor a (2) által generált dinamikai rendszer segítségével a folyamatot az időben matematikailag követni tudjuk, helyesebben előre meg tudjuk mondani, hogyan fog lezajlani. Ehhez csupán a kezdeti állapotot kell ismernünk. Ez a helyzet a klasszikus mechanikában stacionárius, véges sok szabadságfokú rendszerek esetén.

Ha csak a folyamat végállapotát tekintjük lényegesnek és azzal, hogy a jelenség hogyan jut el a végállapotba, nem törődünk, akkor elegendő ismerni a (2) által generált dinamikai rendszer attraktorait. Ekkor a kezdeti állapotról csupán annyit kell tudnunk, hogy melyik attraktor medencéjében helyezkedik el. (Megjegyzendő, hogy a klasszikus mechanika konzervatív és szkleronom rendszereire ez a felfogásmód nem alkalmazható, mivel ilyen rendszernek nincs attraktora.) Érdemes erre az álláspontra helyezkednünk akkor, ha a folyamat kezdeti állapotát nem tudjuk pontosan meghatározni, mert például eleve bizonyos véletlen ingadozásokkal kell számolnunk. Ha a rendszernek kevés számú attraktora van, ezeknek a medencéi nagyok, és a kezdeti állapot bizonytalansága, vagy véletlen ingadozása olyan kicsi, hogy a kezdeti állapotot nem veti ki eredeti medencéjéből, akkor érdemes a dinamikai rendszer determinisztikus modelljével dolgoznunk. Más szóval érdemes a dinamikai rendszer determinisztikus modelljét alkalmazni sztochasztikus folyamatokra is, ha a mondott feltételek teljesülnek, vagyis a folyamat a lényeget tekintve determinisztikus. Ekkor azonban csak az attraktoroknak szabad realitástartalmat tulajdonítanunk, az egyes mozgásoknak és pályáknak nem.

A mondottakat a kockadobás már említett példáján illusztráljuk. A játékkockát légüres térben vízszintes lap fölött dobjuk fel. Ha minket csupán súlypontjának a vízszintes lapjától mért távolsága érdekel, akkor a jelenséget egy szabadságfokúnak tekinthetjük, ami azt jelenti, hogy az állapottér kétdimenziós, az egyik koordináta a súlypont távolsága a vízszintes laptól, a másik a súlypont függőleges irányú sebességkomponense. Bizonyosak lehetünk abban, hogy némi idő eltelte után a kocka a vízszintes lapon nyugalomba kerül, vagyis súlypontjának a vízszintes laptól mért távolsága a fél élhosszal, sebessége zérussal lesz egyenlő. Az állapottér (félélhossz, zérus).pontja tehát a rendszer egyetlen attraktora (feltéve persze, hogy a kockát elég kis sebességgel dobjuk fel, pontosabban, hogy ez a sebesség kisebb az első kozmikus sebességnél, és a vízszintes lap elég nagy).

Ha a kérdés az, hogy hanyast dobunk a kockával, akkor már nem elegendő a súlypont mozgásával foglalkozni, hanem a kockát, mint merev testet kell kezelni. Ekkor a szabadságfokok száma 6, az állapottér dimenziója 12. A vízszintes lap

pontjait ekvivalenseknek tekintve és egymással azonosítva, továbbá a kockának függőleges tengely körüli elforgatását sem tekintve az állapot megváltozásának, bizonyos idő eltelte után a feldobott kocka a vízszintes lapon nyugalomba kerül, sebessége zérus, súlypontjának távolsága a laptól fél élhossz lesz, és az egyes, kettes,... ..., hatos lapok közül pontosan az egyik lesz felül. Ezek szerint ebben az esetben a kocka mozgását leíró dinamikai rendszernek 6 attraktora van, melyek medencéi a kocka szimmetriája miatt a 12 dimenziós állapottérben egyforma mértékűek. Az eldobás pillanatában a kocka kezdeti helyzetének, sebességének, szögsebességének minimális megváltoztatása már áthelyezi a kezdeti értéket az egyik attraktor medencéjéből a másikéba. Ezért kell a kockadobás eredményét valószínűségi változóként kezelnünk.

A kockadobás példájával érdemes szembeállítani a szimultán tanulásra (több tárgy huzamos időn át történő tanulására) a szerző által a [3] dolgozatban javasolt matematikai modellt. Ebben a modellben a tanulás intenzitását leíró dinamikai rendszernek egyetlen attraktora van, és mivel a modell lineáris, ennek medencéje az egész állapottér. Így, bár a tanulás kezdeti intenzitását számos véletlen tényező befolyásolhatja, a tartósan kialakuló stabilis intenzitás egyértelműen megadható.

3. Strukturális stabilitás

Az előző pontban a vizsgált folyamatot leíró dinamikai rendszert egyszer s mindenkorra adottnak vettük, ill. úgy állítottuk be a dolgot, mintha a véletlen hatások csupán a kezdeti értékeket befolyásolhatnák. Ez nem a teljes igazság.

A valóságban változhatnak a dinamikai rendszert meghatározó paraméterek. Állandóan hatnak a mozgásra azok a lényegtelennek tekintett tényezők, melyeket a folyamatot leíró absztrakt modell megalkotásánál elhanyagoltunk, illetve sztochasztikusan változhatnak a mozgás körülményei. Ha egy a valóságban megfigyelhető és a lényeget tekintve determinisztikus folyamatot modellezünk dinamikai rendszerrel, akkor meg kell követelnünk azt, hogy a rendszer az állandóan ható, kis, zavaró körülményekkel, kis *perturbációkkal* szemben „érzéketlen" legyen. Ezt a követelményt egzakt matematikai formába lehet önteni. Itt megelégszünk a szemléletes leírással.

Két dinamikai rendszert *ekvivalensnek* nevezünk, ha az R^n állapottérnek van olyan *homeomorfizmusa* (önmagára való, egy-egyértelmű és mindkét irányban folytonos leképezése), mely az egyik rendszer pályáit a másik pályáiba viszi át.

A továbbiakban feltételezzük, hogy a dinamikai rendszert egy (2) típusú autonóm differenciálegyenlet-rendszer, vagyis egy f sebességmező definiálja. Két dinamikai rendszert *közelinek* nevezünk, ha sebességmezőik és azok deriváltjai egymástól való eltérésének maximuma is kicsi.

Egy dinamikai rendszert *strukturálisan stabilisnak* nevezünk, ha minden hozzá elég közeli dinamikai rendszerrel ekvivalens. A strukturális stabilitásnak ez a fogalma olyan rendkívül mélyen fekvő, nehéz problémákat vet fel, melyek vizsgálata napjainkban a matematikai kutatások egyik reflektorfényében álló területe. Nyilvánvaló az, hogy a strukturálisan stabilis rendszerek a dinamikai rendszerek terének nyílt részhalmazát alkotják, hiszen ha egy rendszer strukturálisan stabilis, akkor minden hozzá közeli rendszer is az. A nagy kérdés az, hogy vajon a strukturálisan stabilis rendszerek a dinamikai rendszerek terében mindenütt sűrűn helyezkednek-e

el, vagyis igaz-e az, hogy majdnem minden találomra felírt dinamikai rendszer strukturálisan stabilis. Erre a kérdésre csupán kétdimenziós, kompakt állapottéren (ill. az állapottér kompakt részhalmazára leszűkítetten) értelmezett dinamikai rendszer esetében adódott pozitív válasz [6]. Gyakorlati célokra azonban a strukturális stabilitás fogalmát kevésbé általánosan, kevesebbet követelve is értelmezhetjük. Lerögzíthetjük a dinamikai rendszert meghatározó sebességmező függvénytípusát, perturbáción csupán az adott függvénytípuson belül néhány paraméter megváltozását értve. *Viszonylagosan strukturálisan stabilisnak* nevezzük a rendszert, ha a paraméterek kis megváltozása az eredetivel ekvivalens rendszert eredményez.

A rendszert meghatározó valós paraméterek száma legyen m. A paraméterekből megalkotjuk az $u=(u_1, u_2, \ldots, u_m)\in R^m$ vektort és (2) helyett autonóm differenciálegyenlet-rendszereknek egy m-paraméteres seregét, az

$$\dot{x}=f(x, u) \tag{3}$$

differenciálegyenlet-rendszert vizsgáljuk. Az $u=(u_1, \ldots, u_m)$ változók R^m terét *paramétertérnek* nevezzük, megkülönböztetésül az $x=(x_1, \ldots, x_n)$ változók R^n állapotterétől. Az $x_1, \ldots, x_n$ változókat a rendszer állapotát meghatározó *belső paramétereknek* is nevezzük, szemben az $u_1, \ldots, u_m$ *külső paraméterekkel*. Egy áramló közeg sűrűségét, nyomását, hőmérsékletét, viszkozitását stb. egy adott pontban belső paramétereknek tekinthetjük, magának a pontnak a térbeli koordinátáit pedig külső paramétereknek. Hasonlóan, valamely kémiai reakció vizsgálatánál a reagensek koncentrációja, a hőmérséklet stb. belső paraméterek, a reakció lejátszódásának színteréül szolgáló térrész pontjainak koordinátái a külsők. Ezekben a példákban a paramétertér dimenziója: $m=3$. Mechanikai mozgások leírásánál, ahol a helyváltoztatás jellegének megragadása a cél, a térbeli koordináták játszhatják a belső paraméterek szerepét és a mozgó pontrendszert, vagy merev testet körülvevő közeg viszkozitása, hőmérséklete stb. lehetnek a külső paraméterek.

A továbbiakban elsősorban azzal a kérdéssel foglalkozunk, hogyan változik a rendszer viselkedése a külső paraméterek változtatásakor.

Ha a rendszer „teljesen" determinisztikus, vagyis az egyes kezdeti értékeknek és pályáknak külön-külön számításba vehető valóságtartalmuk van, akkor e kérdést az állapottérben „lokálisan", egyetlen kezedeti érték szempontjából is vizsgálhatjuk. Tételezzük fel, hogy a (3) rendszer akár minden $u^0\in R^m$ paraméterérték mellett viszonylagosan strukturálisan stabilis. Ha az u^0 pontot kicsivel megváltoztatjuk $u^0+\Delta u$-ra, az $\dot{x}=f(x, u^0)$ és az $\dot{x}=f(x, u^0+\Delta u)$ rendszerek ekvivalensek. Feltételezzük, hogy ha $|\Delta u|$ kicsi, akkor a két rendszer pályáit egymásba átvivő homeomorfizmus is „kicsi", amin azt értjük, hogy a leképezésnél minden képpont távolsága a saját „tárgypontjától" kicsi, vagy másképpen azt, hogy a Δu-hoz tartozó homeomorfizmus a minden pontot helyben hagyó identitás leképezéshez tart $\Delta u\to 0$ esetén. Helyezkedjünk el az x^0 kezdeti állapotban és szemléljük innen, hogy mi történik a külső paraméterek változása esetén. A tipikus esetben, ha a kiinduló külső paraméterérték u^0, az x^0 pont benne van az $\dot{x}=f(x, u^0)$ rendszer $A(u^0)$ attraktorának $M(A(u^0))$ medencéjében. Ha az u pontot a paramétertérben elkezdjük lassan változtatni, akkor a pályák, az $A(u)$ attraktor és ennek $M(A(u))$ medencéje kissé deformálódnak, de mivel $M(A(u^0))$ nyílt halmaz, ha $|\Delta u|=|u-u^0|$ elég kicsi, x^0 még mindig eleme marad az $M(A(u))$ medencének. Az x^0 kezdeti állapotban levő rendszer viselkedése egészen addig nem változik lényegesen, ameddig a medence deformációja nem válik olyan naggyá, hogy az $M(A(u))$ medencét határoló szeperatix

eléri az x^0 pontot. Ekkor az x^0 kezdeti állapotban levő rendszer viselkedése kiszámíthatatlanná válik, majd ha u tovább változik, x^0 átkerülhet egy másik attraktor medencéjébe, és az x^0 kezdeti állapotban levő rendszer viselkedése ugrásszerűen megváltozik.

Minket elsősorban csak a lényeget tekintve determinisztikus rendszerek érdekelnek, amelyeknél az egyedi kezdeti feltételeknek alig van jelentősége. Ezért az előző bekezdésben vázolt jelenséggel, amikor is a rendszer viselkedése a kezdeti érték függvényében, lokálisan változik csak meg ugrásszerűen a külső paraméterek folytonos változtatása során, nem foglalkozunk részletesebben. Ezzel szemben részletesen vizsgáljuk azt a kérdést, hogyan változik a rendszer viselkedése kezdeti értéktől függetlenül, globálisan, a külső paraméterek változása során.

Az R^m paramétertér u pontját *közönséges pontnak* nevezzük, ha ezen u érték mellet a (3) rendszernek egyetlen attraktora van (ennek medencéje az egész állapottér, ill. a tér egész szóba jövő része), és a rendszer viszonylagosan strukturálisan stabilis. Tágabb értelemben közönséges pontnak nevezzük az u pontot akkor is, ha ebben az u pontban a (3) rendszernek ugyan több attraktora van, ezek közül azonban az egyik dominál. *Uralkodónak (dominálónak)* nevezünk egy attraktort például akkor, ha medencéjének mértéke lényegesen nagyobb, mint bármelyik másik attraktoré. Később, fontos speciális esetekben másképpen is értelmezzük egy attraktor uralmát a többi fölött. Közönséges pontban a dinamikai rendszer a kezdeti értéktől függetlenül egyértelműen meghatározott állapotban van (pontosabban rövid idő alatt ebbe az állapotba kerül), és ez igaz a kezdeti értékek nagy részére, a tipikus esetben akkor is, ha a pont csak tágabb értelemben közönséges. Továbbá, ha ilyen pontból indulunk ki, és a körülmények, a külső paraméterek kissé megváltoznak, a rendszer állapota, belső dinamikája nem változik lényegesen.

Az R^m paramétertér u pontját *egyszerű katasztrófapontnak* nevezzük, ha abban a (3) rendszernek két, vagy több nagyjából egyenlő erősségű, domináló attraktora van, és a rendszer viszonylagosan strukturálisan stabilis. Egyszerű katasztrófapont környezetében is minden pont egyszerű katasztrófapont; a két (vagy több) egymással vetélkedő attraktor ilyenkor strukturálisan stabilisan van jelen. Egyszerű katasztrófa — pontban azt mondjuk, hogy a rendszer a *konfliktus* állapotában van. A rendszer ilyenkor az egyik attraktor által jellemzett állapotból a másikba ugorhat attól függően, hogy a kezdeti érték hogyan változik.

Az R^m paramétertér u pontját *lényeges katasztrófapontnak* nevezzük, ha ebben a pontban a (3) rendszer (és különösen az attraktora, vagy attraktorai) viszonylagosan nem strukturálisan stabilis. Lényeges katasztrófapontban egyrészt a rendszer állapota a külső paraméterek kis megváltozása mellett ugrásszerűen megváltozhat, ilyenkor ugyanis eltűnhet egy korábban domináló attraktor és tőle távol keletkezhet egy vagy több másik. Ekkor azt mondjuk, hogy a rendszer az *ugrás* állapotában van. Másrészt előfordulhat az, hogy a külső paraméterek kis megváltozása következtében az egyetlen uralkodó attraktor kettéválik és a rendszer sima átmenettel az egyértelmű meghatározottság állapotából a konfliktus állapotába kerül. Ez utóbbi esetben azt mondjuk, hogy a lényeges katasztrófapont a rendszer *elágazási (bifurkáció) pontja* és a rendszer itt *elágazik (bifurkálódik)*.

Ha az előbbiekben mondottakat valóságos rendszerek modellezésére kívánjuk felhasználni, figyelembe kell vennünk, hogy a valóságos rendszerek az esetek nagy részében „emlékeznek", más szóval nem közömbösek az iránt, hogy a külső paraméterek milyen előzmények után, milyen pályán kerültek a paramétertér adott

pontjába. Ezt a tulajdonságot fejezi ki a *késlekedés törvénye,* melynek lényege a következő. Ha a paramétertér egy közönséges pontjából indulunk ki és meghatározott irányban haladva átmegyünk egy lényeges katasztrófaponton, akkor a tipikusan jellemző esetben a korábban uralkodó egyetlen attraktor mellett megjelenik egy újabb, lényegesen gyengébb attraktor. Ha tovább haladunk az eredeti irányban, akkor az új attraktor erősödik (például medencéje terebélyesedik), a régi, domináló attraktor pedig gyengül, míg eljutunk egy olyan pontba, melyben a két attraktor egyenlő erősségűvé válik, vagyis a rendszer a konfliktus állapotába kerül. Még tovább haladva az új attraktor válik dominálóvá és a régi gyengébbé, *a rendszer azonban nem ugrik át azonnal az új attraktor által jellemzett állapotba.* Ez az ugrás csak késve, a külső paraméterek további azonos irányú változása után következik be, esetleg csupán akkor, amikor a régi attraktor teljesen eltűnik.

4. Thom katasztrófaelmélete

Abban az általánosságban, ahogy az eddigiekben vázoltuk a dinamikai rendszer, az attraktorok, a strukturális stabilitás fogalmát, osztályoztuk a katsztrófákat, az elmélet számos rendkívül nehéz problémát vet fel. Ezen a területen, mely a mai a matematika egyik legdinamikusabban fejlődő fejezete, a nyitott kérdések száma jóval nagyobb, mint a lezártaké. RENÉ THOM 1970 körül alkotta meg a „katasztrófaelméletet", melynek első rendszerességre törekvő kifejtése 1972-ben megjelent [10] könyvében található. A katasztrófaelmélet célja, hogy matematikai módszert adjon olyan valóságos jelenségek leírására, melyek során a paraméterek kis megváltozása a jelenségben ugrásszerű, minőségi változást eredményez. A maximális cél az, hogy matematikailag jellemezni, osztályozni lehessen az összes előforduló „katasztrófákat". Bár az elméletnek ettől a céljától még nagyon messze vagyunk, az alkalmazások máris tömegesen jelentkeznek olyan egymástól távol álló tudományterületeken, mint a fizika, a rugalmasságtan, a hullámtan, a kémia, a biológia, az embriológia, a paleontológia, a geológia, a pszichológia, a nyelvtudomány, a szociológia stb.

Egy fontos, speciális esetben azonban már maga THOM elvégezte a teljes osztályozást, bár erre vonatkozó alaptételét (helyesebben sejtését) csak néhány évvel később sikerült bebizonyítani (lásd [11]). Ezt a speciális esetet mostanában *elemi katasztrófaelméletnek* szokták nevezni; a továbbiakban ezzel foglalkozunk. Feltételezzük, hogy a (3) rendszer ún. *gradiens-rendszer,* ami azt jelenti, hogy van olyan $V(x, u)$ valós értékű *potenciálfüggvény,* melynek az f függvény negatív gradiense (a belső paraméterek szerint):

$$f(x, u) = -\operatorname{grad}_x V(x, u), \tag{4}$$

vagyis ha f koordinátáit $(f_1, f_2, \ldots, f_n)$-nel jelöljük,

$$f_i = -V'_{x_i}, \quad (i = 1, 2, \ldots, n).$$

Ezek szerint dinamikai rendszerünk most az

$$\dot{x} = -\operatorname{grad}_x V(x, u) \tag{5}$$

alakot veszi fel. Feltételezzük továbbá, hogy (5) attraktorainak halmaza csupán véges sok izoláltan elhelyezkedő egyensúlyi helyzetből áll. Az (5) rendszer egyen-

súly helyzetei megegyeznek a V függvény stacionárius pontjaival (azokkal a pontokkal, melyekben V összes x_i szerinti elsőrendű parciális deriváltja zérus). Miután (5) szerint a mozgás sebességvektora mindig V csökkenésének irányába mutat, az attraktorok V minimumhelyei az R^n térben. Gradiens-rendszernél általában azt az attraktort nevezzük *uralkodónak (dominálónak)*, amelyben a potenciálfüggvény értéke kisebb, mint az összes többi minimumhelyen. Ilyen rendszer akkor van a konfliktus állapotában, ha ezt a „minimális minimumértéket" a potenciálfüggvény több helyen veszi fel.

E feltevések következtében a (3), illetve most már az (5) rendszer és az ő attraktorai strukturális stabilitásának kérdése a skalárértékű V függvény és az ő minimumhelyei (viszonylagos strukturális) stabilitásának kérdésére redukálódik.

Arról van szó, hogy rögzített u mellett az n-változós V függvénynek véges sok minimumhelye van, és a kérdés az, mit csinálnak ezek a minimumhelyek ha u értékét kicsit megváltoztatjuk. Lehetséges az, hogy u kis megváltoztatása esetén ezek a minimumhelyek kissé megváltoztatják helyzetüket, és a minimális függvényértékek is megváltoznak kissé, de a minimumhelyek száma nem változik. Az is lehetséges azonban, hogy u kis megváltoztatása minimumhelyek eltűnését, összeolvadását, vagy újak keletkezését vonja maga után. Ahhoz, hogy ezt a kérdést tanulmányozhassuk és a potenciálfüggvények lehetséges viselkedéseit osztályozhassuk, néhány fogalomra van szükség.

Differenciálható sokaságnak nevezzük a közönséges kétdimenziós, sima felület magasabb dimenziós megfelelőjét magasabb dimenziós térben. Olyan leképezésekről kell beszélnünk, melyek egy differenciálható sokaságot egy másik differenciálható sokaságba képeznek le. Például egy gömbfelületet egy síkba képez le az a leképezés, melynek során a gömbfelület minden egyes pontját merőlegesen rávetítjük a síkra. Legyenek v és w egy M differenciálható sokaság folytonosan differenciálható leképezései egy N differenciálható sokaságba és m, ill. p az M sokaság pontjai. Azt mondjuk, hogy a v leképezés az m pontban *lokálisan ekvivalens* a w leképezéssel a p pontban, ha van m-nek olyan U_m és p-nek olyan U_p környezete, továbbá U_m-nek U_p-re való g és N-nek N-re való h *diffeomorfizmusa* (mindkét irányban folytonosan differenciálható homeomorfizmusa), amelyekre az

$$\begin{array}{ccc} U_m & \xrightarrow{v} & N \\ {\scriptstyle g}\downarrow & & \downarrow{\scriptstyle h} \\ U_p & \xrightarrow[w]{} & N \end{array}$$

diagramm kommutatív, vagyis amelyekkel a következő összetett leképezésekre fennáll $h \circ v = w \circ g$, és $g(m) = p$. Az m pontot a v leképezés *szinguláris pontjának* nevezzük, ha m-nek nincs olyan környezete, melynek v diffeomorfizmusa lenne. Ha m, ill. p a v, ill. w leképezés szinguláris pontja és v az m pontban lokálisan ekvivalens w-vel a p pontban, akkor azt mondjuk, hogy az m és a p *szinguláris pontok* ekvivalensek, vagy ugyanabban a *topológiai szingularitás típusba* tartoznak.

Visszatérve az elemi katasztrófaelmélet feltételeit kielégítő (5) rendszerhez, jelöljük M_V-vel a V függvény stacionárius pontjainak halmazát az $R^n \times R^m$ térben. M_V tehát azoknak az (x, u) pontoknak a halmaza, amelyekben $f(x, u) =$

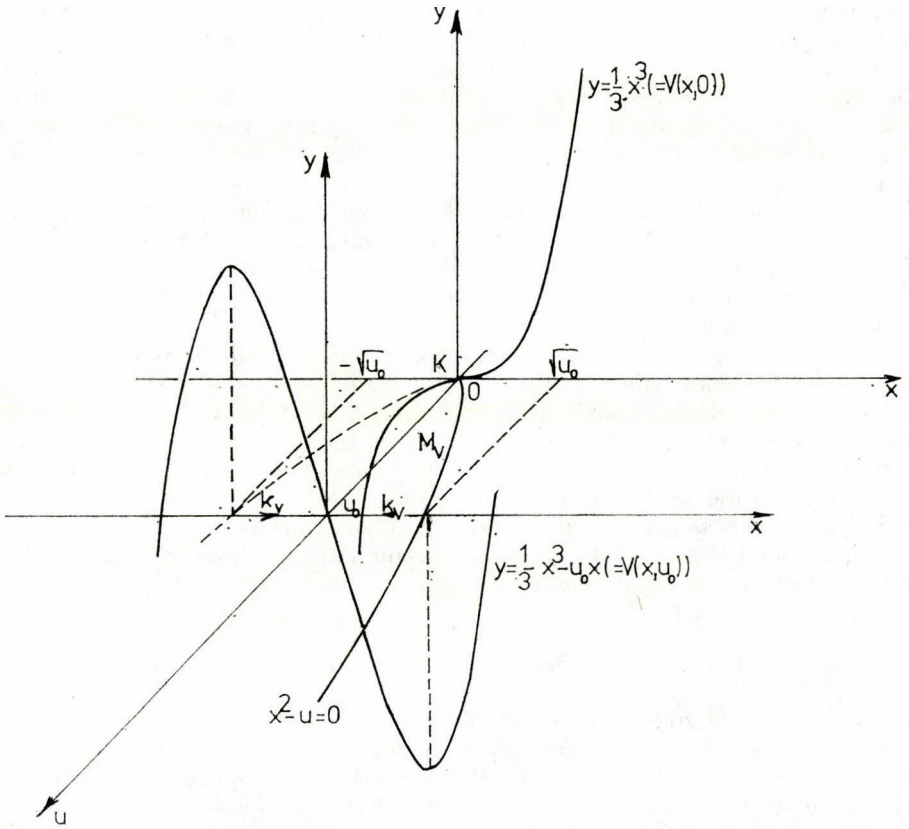
$-\text{grad}_x V(x, u)=0$. Az M_V-t meghatározó feltétel valójában n számú egyenletet jelent az $n+m$ számú ismeretlenre:

$$V'_{x_1}(x, u) = 0, \ldots, V'_{x_n}(x, u) = 0,$$

$x=(x_1, \ldots, x_n)$, $u=(u_1, \ldots, u_m)$. A tipikus esetben azt várjuk, hogy ezekből az egyenletekből legalábbis lokálisan n számú ismeretlent, éspedig az $(x_1, \ldots, x_n)$-eket kifejezhetjük, mint a többi ismeretlen $(u_1, \ldots, u_m)$ kellően jó tulajdonságokkal rendelkező függvényeit. Az M_V halmaz azért fontos számunkra, mivel ebben helyezkednek el a V függvény minimum helyeinek, vagyis az (5) rendszer attraktorainak megfelelő pontok. Ha egy u pont valamely környezetében M_V egyenlete megadható $x_1=$ $=x_1(u), \ldots, x_n=x_n(u)$ alakban, ez azt jelenti, hogy ezekhez az u értékekhez V-nek egyetlen stacionárius pontja, vagyis az (5) rendszernek legföljebb egyetlen attraktora tartozik. Ezek az u pontok tehát az R^m paramétertér közönséges pontjai. Más u pontok környezetében az M_V halmaz végesszámú rétegre bontható fel, melyek különkülön megadhatók más-más $x_1=x_1(u), \ldots, x_n=x_n(u)$ egyenletrendszerrel. Ezek azok az u értékek, melyek mellett az (5) rendszernek egynél több attraktora lehet, ezek között lehetnek tehát az egyszerű katasztrófapontok. Végül azok az u pontok, melyek környezetében M_V nem képzelhető el az R^m paramétertér fölötti egyszerű felületdarabként és nem is rétegezhető úgy, hogy egy-egy réteg ilyen felületdarab legyen, a lényeges katasztrófapontok. Ezek azok a pontok, melyekben M_V „visszahajlik", „meggyűrődik" stb.

Az előbb mondottak matematikai leírását annak a leképezésnek a vizsgálata teszi lehetővé, mely az M_V halmazt az R^m paramétertérbe vetíti le. A V potenciál függvényhez tartozó *katasztrófa-leképezésnek* nevezzük és k_V-vel jelöljük azt a függvényt, mely az M_V halmazon van értelmezve és M_V minden (x, u) pontjához hozzárendeli az R^m paramétertér u pontját: $k_V(x, u)=u$, $(x, u)\in M_V$. Az R^m térnek azok a pontjai, melyeknek van olyan környezetük, hogy a k_V leképezés diffeomorfizmus e környezet és az M_V halmaz egy nyílt darabja, ill. az M_V halmaz egy rétegének nyílt darabja között, a közönséges pontok, ill. az egyszerű katasztrófapontok. A k_V leképezés szinguláris pontjainak képpontjai a lényeges katasztrófapontok. A paramétertér ezen pontjaiban a V potenciálfüggvény jellege minőségileg megváltozik. A lényeges katasztrófapontok K halmazát vagyis a k_V leképezés szinguláris pontjainak vetületét az R^m paramétertérben *katasztrófahalmaznak* nevezzük.

Az előbb mondottakat egy egyszerű példán illusztráljuk (lásd 1. ábra). Az $\dot{x}=-x^2+u$ dinamikai rendszernek negatív u-kra nincs egyensúlyi helyzete, pozitív u paraméterérték esetén két egyensúlyi helyzete van $x=\sqrt{u}$, ill. $x=-\sqrt{u}$ és megoldásai az $x(t)=\sqrt{u}\,\text{th}\,(t\sqrt{u}+c)$, ill. $x(t)=\sqrt{u}\,\text{cth}\,(t\sqrt{u}+c)$ függvények, melyek $\sqrt{u}$-hoz tartanak, ha t végtelenhez tart. Tehát $x=\sqrt{u}$ a rendszer egyetlen attraktora, bár medencéje nem az egész x egyenes, hanem annak csupán az $x>-\sqrt{u}$ egyenlőtlenséggel jellemzett fele. A rendszer jobb oldala a $V(x, u)=1/3x^3-ux$ potenciál negatív gradiense, $V'_x=x^2-u$ és V-nek az $x=\sqrt{u}$ pont valóban az egyetlen minimum-helye. V stacionárius pontjainak M_V halmaza az (x, u) síkban az $x^2-u=0$ egyenletű parabola. E parabolának az $x>0$ egyenlőtlenséggel jellemzett ága („rétege") felel meg attraktoroknak, a másik ág a V függvény maximum-helyeinek felel meg. A k_V leképezés a parabolát az u-tengelybe (mely most az egydimenziós paramétertér) vetíti; a pozitív u helyek közönséges pontok. A K katasztrófahalmaz



1. ábra

egyetlen pontból, az $u=0$ lényeges katasztrófapontból áll. Ha ezen a ponton negatív irányban haladunk át, a rendszer attraktora eltűnik, vagyis a rendszer minőségileg megváltozik.

Jelöljük C^∞-lel a $R^n \times R^m$ téren értelmezett, valós értékű, akárhányszor folytonosan differenciálható függvények osztályát. A C^∞ függvényosztálynak van egy részhalmaza, melyet a *tipikus függvények* osztályának nevezünk. A tipikus függvényeket úgy képzelhetjük magunk elé, hogy sem ők, sem pedig az x_i változók szerinti parciális deriváltjaik nem azonosan állandók, bár a pontos definíció ennél lényegesen bonyolultabb és valamivel tágabb.

Thom tétele, mely az elemi katasztrófaelmélet alapját képezi, a következőket mondja. Ha a külső paraméterek száma, m nem nagyobb 5-nél, akkor a tipikus függvények halmaza C^∞-ben mindenütt sűrű és nyílt. (Ez valójában azt jelenti, hogy majdnem minden végtelen sokszor folytonosan differenciálható függvény tipikus.) Ha V tipikus, akkor stacionárius pontjainak M_V halmaza m-dimenziós differenciálható sokaság. (Ez azért fontos, mert az állapottér n dimenziója pl. biológiai, pszichológiai, vagy gazdasági folyamatok esetén igen nagy lehet. Az M_V halmaz dimen-

ziója azonban ettől nem függ.) Továbbá ekkor a k_V katasztrófa-leképezés minden egyes szinguláris pontja véges sok ismert szingularitás-típus egyikébe tartozik. Pontosabban, ha m értéke 1, 2, 3, 4, ill. 5, akkor van pontosan 1, 2, 5, 7, ill. 11 különböző típusú szinguláris pont úgy, hogy minden tipikus függvény katasztrófa leképezésének minden egyes szinguláris pontja ezek egyikével ekvivalens. A szinguláris pontoknak ezeket az alaptípusait nevezzük *elemi katasztrófáknak*. Végül a k_V katasztrófa-leképezés értelmezési tartományának minden pontjában lokálisan stabilis a V függvény kis megváltoztatásával szemben. (Ezt úgy kell érteni, hogy a k_V és k_W katasztrófa-leképezéseket közelinek tekintjük, ha V és W közeli tipikus függvények, és k_V lokálisan ekvivalens minden hozzá elég közeli k_W-vel. Különösen fontos az, hogy ez természetesen érvényes k_V szinguláris pontjaira is, vagyis ezek strukturálisan stabilisak.)

THOM előbbiekben ismertetett tételének jelentősége valóságos folyamatok tudományos leírásában felmérhetetlen. E tétel szerint ui. bármilyen (fizikai, kémiai, biológiai, pszichológiai, gazdasági, társadalmi stb.) folyamatot modellezünk is gradiens-rendszerrel úgy, hogy a lényeges külső paraméterek számát sikerül 5 alá szorítanunk, a folyamat alakulásában elsőrendűen fontos jelenség, a strukturális stabilitás elvesztése véges sok THOM által osztályozott és többé-kevésbé szemléltethető típus egyike szerint történik. Továbbá a strukturális stabilitás elvesztésének típusai, az elemi katasztrófák maguk is struktúrálisan stabilisak és ezért a valóságban, a természetben megfigyelhetők.

5. A csúcs-katasztrófa

Az elemi katasztrófák THOM-féle osztályozásának és leírásának ismertetésétől itt el kell, hogy tekintsünk (lásd [9] és [10]).

Azt az esetet, amikor egyetlen külső paraméter van az előző pont példájában már bemutattuk. Azzal az esettel, amikor a független külső paraméterek száma $m=2$, részletesen foglalkozunk, mivel ez több okból lényeges. Először is szemben az $m=1$ esettel ez már valóban érdekes. „Erényes mindenki magában is lehet, a bűnhöz mindig kettő kell" írja HEINE, márpedig a közfelfogás azt tartja, hogy a bűn érdekesebb, mint az erény. Másrészt az $m=2$ esetben a stacionárius pontok M_V halmaza kétdimenziós sokaság. Ezért a belső paraméterek n számától (az állapottér dimenziójától) függetlenül M_V a közönséges háromdimenziós tér egy sima felületével reprezentálható. Ha $m>2$, akkor az M_V sokaságot már nem tudjuk igazán magunk elé képzelni.

Két független külső paraméter esetén, THOM tétele szerint két elemi katasztrófa létezik. Ezek közül az egyik az ún. kétdimenziós *ránc* (fold)-*katasztrófa* az előző pont már idézett példájának kétdimenziós analogonja (a parabolának, mint a stacionárus pontok halmazának egy parabolikus henger lép a helyébe). Miután a kétdimenziós ránc-katasztrófa az egydimenzióshoz képest semmi lényeges új vonást nem mutat, ezzel nem foglalkozunk.

A másik kétdimenziós elemi katasztrófát *csúcs* (cusp)-*katasztrófának* nevezzük. A csúcs-katasztrófa rendkívül tanulságos, lényeges és szemléletes. Tanulságos, mivel a konfliktus, az ugrás és az elágazás egyaránt megtalálható benne. Fontos, mivel sok nagyon különböző valóságos folyamat lényegét meg lehet ragadni két más-más irányban ható külső paraméter kiválasztásával és ekkor a jelenség lényege

a csúcs-katasztrófával modellezhető. E pont hátralevő részében ezzel foglalkozunk. A két külső paramétert u-val és v-vel jelöljük, az állapottér dimenzióját az egyszerűség kedvéért egynek vesszük: $n=1$, és a belső paramétert x-szel jelöljük. A potenciálfüggvény

$$V(x, u, v) = \frac{1}{4}x^4 - \frac{1}{2}ux^2 - vx. \tag{6}$$

A V függvény gradiense $V'_x(x, u, v) = x^3 - ux - v$, vagyis a stacionárius pontok M_V halmazát az $R^3 = R \times R^2$ térben az $x^3 - ux - v = 0$ egyenlőség jellemzi. Az M_V felület egy paraméteres előállítása:

$$u = u, \quad v = x^3 - ux,$$

és ez utóbbi egyenletek egyben megadják a k_V katasztrófa-leképezést, mely az M_V felület minden (x, u, v) pontjához hozzárendeli a paramétersík (u, v) pontját. A k_V leképezés szinguláris pontjai azok, amelyekben Jacobi-determinánsa zérus:

$$\frac{\partial(u, v)}{\partial(u, x)} = 3x^2 - u = 0.$$

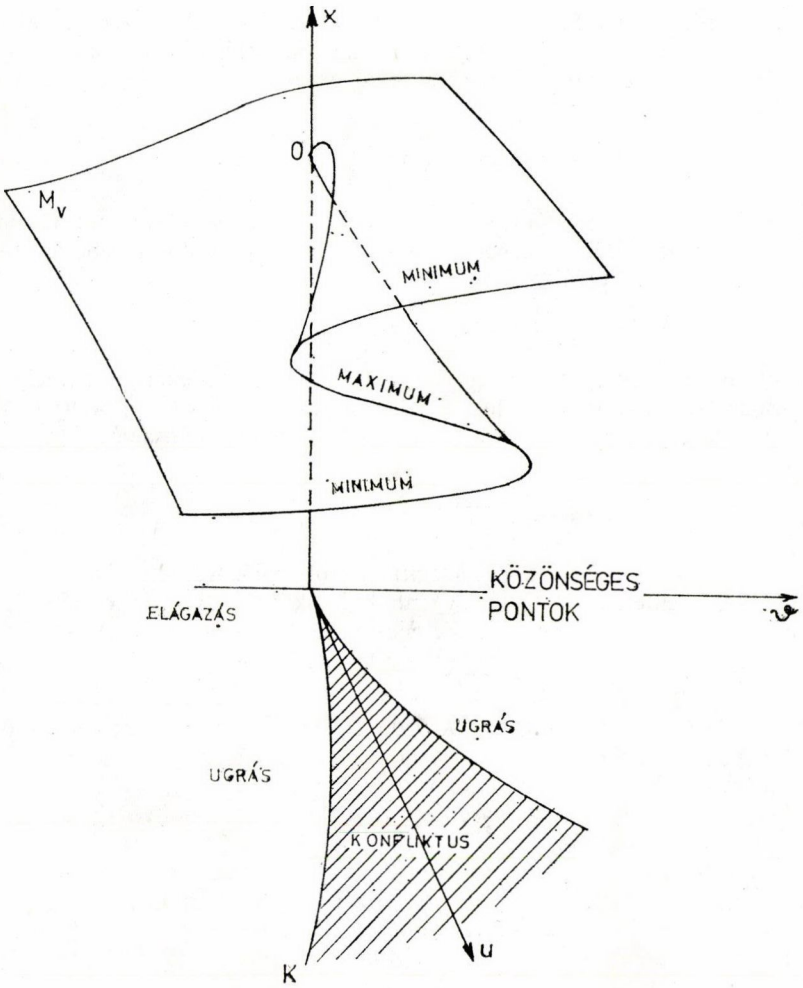
A szinguláris pontok koordinátái kielégítik az utóbbi egyenletet és természetesen az M_V felület egyenletét. Ha e két egyenletből kiküszöböljük az x változót, megkapjuk a szinguláris ponthalmaz vetületének egyenletét:

$$27v^2 - 4u^3 = 0. \tag{7}$$

Ez utóbbi egyenlet jellemzi tehát a K katasztrófa-halmazt a paramétersíkon (lásd 2. ábra). A K katasztrófa-halmaz csúcsos alakjáról nyerte ez az eset a nevét.

A K halmaz az (u, v) paramétersíkot két részre osztja. A „külső" (nem bevonalkázott) rész fölött az M_V felület egyrétűen helyezkedik el. Könnyen belátható, hogy az M_V felületnek a sík ezen részei feletti pontjaiban $V''_{xx}(x, u, v) = 3x^2 - u > 0$. Így ezekben a pontokban a V függvénynek egyetlen minimuma van. Tehát a paramétersíknak a K halmazon „kívüli" pontjai a rendszer közönséges pontjai. A K halmazon „belüli" (bevonalkázott) rész fölött az M_V felület háromrétűen helyezkedik el; az alsó és a felső rétegen levő pontokban V-nek minimuma van, a középső rétegen levő pontokban pedig maximuma. Így a rendszernek a K halmazon „belüli" síkrész pontjaiban két attraktora van. A két attraktor a bevonalkázott síkrész közepe táján egyforma erősségű, tehát e pontok a rendszer egyszerű katasztrófa-pontjai.

A K halmaz pontjaiban a rendszer nem strukturálisan stabilis. Ha a paramétersíkban a K halmaz valamely a csúcstól különböző pontján „kívülről befelé" haladunk át a rendszernek, amelynek addig egyetlen attraktora volt, új, gyenge attraktora keletkezik, tehát a rendszer minőségileg megváltozik. Ha tovább haladunk a belső rész közepe felé, az új attraktor egyre erősödik és a rendszer a konfliktus állapotába kerül. Még tovább haladva az eredeti irányban, most már kifelé, az új attraktor válik dominálóvá, de a késlekedés törvénye szerint a rendszer csak akkor ugrik át szükségszerűen az új attraktor által jellemzett állapotba, amikor „belülről kifelé" haladva átlépjük a K halmazt. A K halmaznak a csúcstól különböző pontjaiban tehát a rendszer az ugrás állapotában van, de az ugrás szükségszerűen csak akkor következik be, ha „belülről kifelé" haladunk.



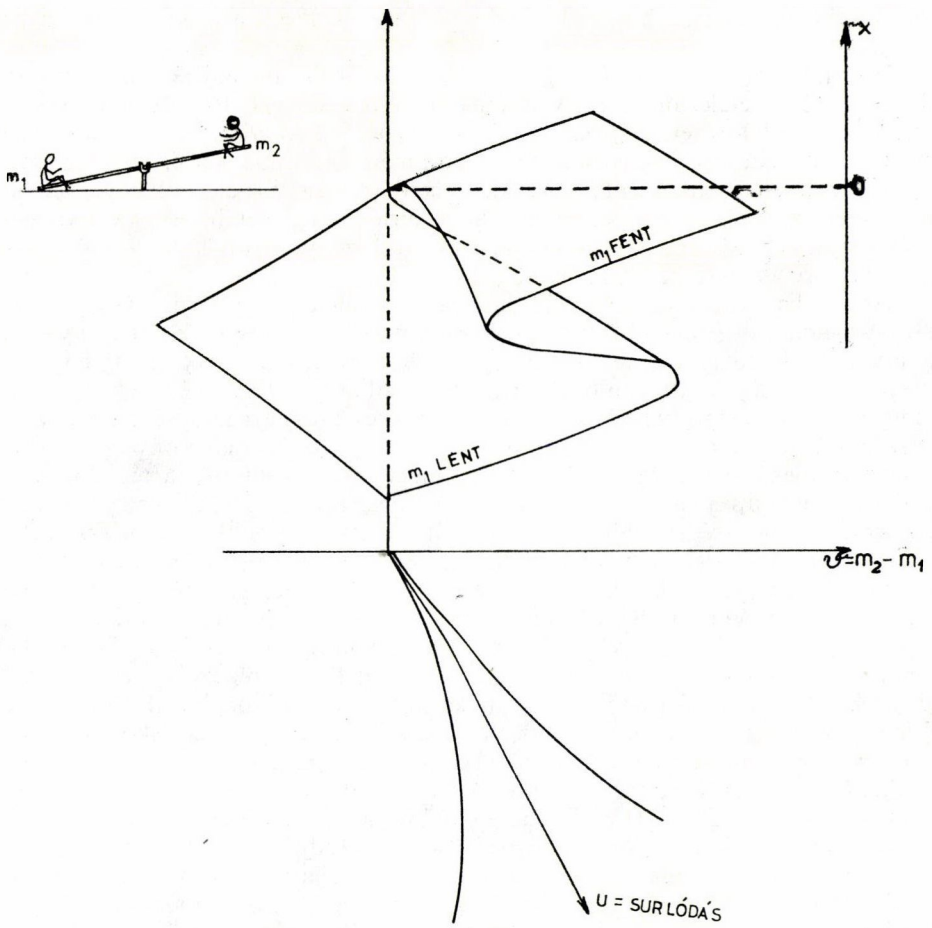
2. ábra

A K halmaz csúcspontjában, ha v értékét zérusban rögzítve az u paraméter értékét a negatív tartományból a pozitívba növeljük, a rendszer egyetlen attraktora kettéválik, két egyenlő erősségű attraktorrá. A v paraméter minimális változtatása pozitív, ill. negatív irányban eldönti, hogy a rendszer az M_V felület felső, vagy alsó rétegére kerül-e, melyik attraktort választja. A csúcsban tehát a rendszer elágazik.

Egyébként a v változót *normál-paraméternek*, az u változót *megosztó* (splitting) *paraméternek* szokták nevezni. Ez utóbbi elnevezés magyarázata az, hogy minél nagyobb u értéke, annál nagyobb a rendszer megosztottsága, ti. a távolság a két attraktor között.

A csúcs-katasztrófát egy egyszerű mechanikai példával szemléltetjük. Képzeljünk el egy m_1 és egy m_2 tömegű gyereket, amint egy egyszerű mérleghintán,

a tengelytől egyforma távolságra ülve hintáznak. Feltételezzük, hogy a hinta nyugalmi (tapadási) súrlódása nagy, különösen a két szélső helyzetben, amikor az egyik gyerek lába a földre ér, és a tengely súrlódásához hozzáadódik a láb-föld tapadási súrlódás, valamint az izmok és ízületek által kifejtett ellenállás (csillapítás). Feltételezzük továbbá, hogy a hinta ütközése a földhöz abszolút rugalmatlan. Anélkül, hogy a rendszer belső dinamikáját elemeznénk, mozgásegyenleteit felírnánk, pusztán tapasztalati, intuitív megfigyelés alapján a jelenséget a csúcs-katasztrófával modellezhetjük. Normál-paraméternek a két tömeg különbségét választjuk: $v=m_2-m_1$, az u megosztó paraméter a súrlódás. A rendszer állapotát jellemző x belső paraméter jelentése: az a kezdő impulzus, mely a hinta vízszintes helyzetének eléréséhez szükséges a súrlódási veszteség figyelembevételével (függetlenül attól, hogy ezután mi



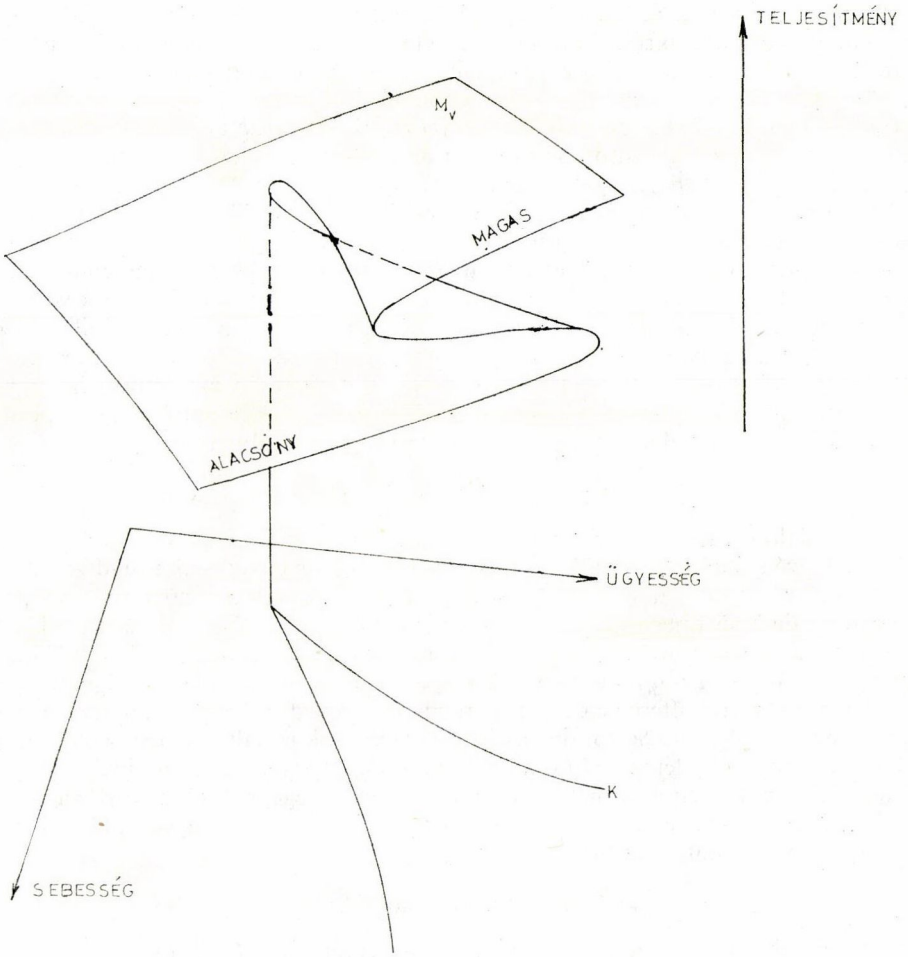
3. ábra

történik). Ezt az impulzust pozitívnak vesszük, ha az m_1 tömeget lefelé kell mozgatni és negatívnak, ha fölfelé (lásd 3. ábra).

Ha nincs súrlódás, vagy minimális, akkor a hintának egyetlen attraktora van: a nehezebb gyerek van a földön. Ha a súrlódás jelentős, és eredetileg m_1 lényegesen nagyobb mint m_2, akkor m_1 van lent. Ha most az m_2 tömeget növeljük és az eléri, majd túlszárnyalja m_1-et, a hinta még nem vált át. Ez az átváltás csak akkor következik be, miután m_2 eléri azt az értéket, mely elég mind az m_1 tömeg felemeléséhez, mind pedig a súrlódási ellenállás legyőzéséhez. A szükséges m_2-m_1 különbség és az ugrás annál nagyobb, minél nagyobb a súrlódás. Ha az átváltás már megtörtént egy m_2-m_1 értéknél, a visszaváltás nem fog bekövetkezni ugyanitt. A visszaváltáshoz m_2-t jobban le kell csökkenteni, ill. m_1-et növelni. Ha m_1 és m_2 közelítőleg egyenlő és a súrlódást a zérus értékről növeljük, a rendszer egyetlen attraktorának a helyébe kettő lép, vagyis a rendszer elágazik. Ha az m_2 és m_1 tömegek eltérése kicsi és a súrlódás megfelelően nagy, a hintának mind a két szélső helyzete attraktor. Ekkor lehet hintázni.

Felhívjuk a figyelmet arra, hogy a csúcs-katasztrófa modelljének alkalmazásánál nem kell feltételeznünk, hogy a dinamikai rendszernek (6) alakú potenciálfüggvénye van. THOM tétele éppen azt biztosítja, hogy gyakorlatilag minden kellően sima gradiens-rendszer, melyben a külső paraméterek száma kettő, vagy a ránc-, vagy pedig a csúcs-katasztrófával modellezhető. Az előző példával kapcsolatban felmerülhet az aggály, hogy vajon jogos-e a súrlódás figyelembe vételével vizsgált hintát gradiens-rendszerként kezelnünk. A következő pontban látni fogjuk, hogy a „gradiens-rendszer feltételt" enyhíteni lehet.

A csúcs-katasztrófa alkalmazására még egy példát tárgyalunk (lásd ZEEMAN [12]). A példa a *„lassan járj, tovább érsz"* közmondás, melyre ZEEMAN nyomán egy munkalélektanilag érdekesnek tűnő modellt adunk. Vizsgáljuk egy dolgozó teljesítményének alakulását, pontosabban x-szel jelöljük és teljesítménynek nevezzük a dolgozó (mennyiségi) teljesítményszázalékának és munkája minőségének valamiféle szorzatát (ami alacsony akkor is, ha teljesítményszázaléka alacsony és akkor is, ha selejtszázaléka nagy). Alapvető külső paramétereknek tekintjük a munka sebességét (amit például szalagon dolgozó munkás esetében a szalag mozgása határoz meg), ill. a dolgozó ügyességét (ebbe beleértjük rátermettségét, szakképzettségét, begyakorlottságát). A teljesítmény alakulását sok más külső, egyebek között véletlen tényező befolyásolja; ezek közül azonban a megkívánt sebességet és az ügyességet tekintjük lényegeseknek. Ebben az esetben a teljesítmény alakulását a csúcs-katasztrófa modellezi (lásd 4. ábra). A két külső paraméter tengelye most az előző ábrákkal szemben a csúcshoz képest 45°-kal el van forgatva. Ha a dolgozó ügyessége kicsi, és kis sebességet követelnek tőle, akkor teljesítménye egy viszonylag alacsony értékkel jellemezhető. Ha ügyessége változatlan marad, de a sebességet fokozzuk, teljesítménye csökkenni (selejtszázaléka növekedni) fog. A teljesítményt legjobban úgy növelhetjük, ha a dolgozó ügyességét és a megkívánt sebességet egyszerre növeljük vigyázva arra, hogy „az ügyesség mindig előbbre tartson". Ha a dolgozó ügyessége viszonylag magas, állandó érték és a sebességet elkezdjük növelni, a teljesítmény egy bizonyos ponton ugrásszerűen lezuhan. (Gondoljunk például egy gépkocsivezetőre, akitől egyre nagyobb sebességet és közben a közlekedési szabályok betartását követeljük). Fordítva, ha a sebességet változatlanul tartva az ügyesség növekszik, egy bizonyos ponton a teljesítmény ugrásszerűen megnő. (Gondoljunk egy tanuló gépkocsivezetőre.) Ha a dolgozó ügyessége nagy, és nagy sebességet követel-



4. ábra

nek tőle, (a katasztrófa-halmaz által határolt csúcsos síktartomány közepe táján) a dolgozó teljesítménye, vagy magas lesz, vagy alacsony, attól függően, hogy „milyen napja van".

A csúcs-katasztrófa további biológiai, pszichológiai, szociológiai, gazdasági alkalmazásaira nézve lásd [12], [2].

6. Ljapunov-függvények

Az elemi katasztrófaelméletnek az a követelése, hogy a dinamikai rendszert generáló autonóm differenciálegyenlet-rendszer jobb oldala egy potenciálfüggvény gradiense legyen, feleslegesen és túlzottan leszűkíti az elmélet hatókörét. Ehhez a feltevéshez ragaszkodva, például az áramlástanban szükségképpen le kellene monda-

nunk mindazokról az áramlásokról, melyeknek nincs sebességpotenciáljuk, vagyis az örvényes áramlásokról. Márpedig a réteges áramlások átalakulása turbulens áramlásokká kifejezetten a katasztrófaelmélet hatókörébe tartozó jelenség.

A „gradiens-rendszer feltevéstől" azonban viszonylag könnyen megszabadulhatunk. LJAPUNOV 1892-ben dolgozta ki azt a róla elnevezett elméletet (lásd pl. [1]), mely ma is az egyik legjelentősebb módszer differenciálegyenlet-rendszerek egyensúlyi helyzetei stabilitásának vizsgálatában. A „Ljapunov-féle direkt módszer" egy speciális esetre kimondott, tipikus tétele a következő. Ha az $\dot{x}=f(x)$ (2) differenciálegyenlet-rendszernek $x\equiv a$ egyensúlyi helyzete, és van olyan $V(x)$ skalárfüggvény, mely az a helyen zérus, a-nak egy környezetében (a kivételével) mindenütt pozitív és melynek a rendszerre vonatkozó deriváltja ebben a környezetben (a kivételével) mindenütt negatív, akkor az a egyensúlyi helyzet aszimptotikusan stabilis, vagyis attraktor (és a szóban forgó környezet mindenesetre beletartozik a medencéjébe). Itt a V függvénynek a (2) rendszerre vonatkozó deriváltján azt értjük, hogy V-be behelyettesítjük a differenciálegyenlet-rendszernek az állapottér adott pontján áthaladó pályájú megoldását, és ezt az összetett függvényt differenciáljuk:

$$\dot{V}_{(2)}(x)=\dot{V}(x(t))=\dot{x}\cdot\operatorname{grad}V=f(x)\cdot\operatorname{grad}V(x)$$

(a jobb oldalon a két vektor skaláris szorzata áll). A V függvényt a rendszer „Ljapunov-függvényének" nevezzük. Általában egy differenciálegyenlet-rendszer *Ljapunov-függvényének* nevezünk egy skalárfüggvényt, ha annak a rendszerre vonatkozó deriváltja mindenütt negatív, vagy legalábbis nem pozitív. Az előbb leírt tétel szemléletes jelentése az, hogy a rendszer pályái mentén a Ljapunov-függvény csökken és így ha a e függvény minimumhelye, akkor a rendszernek attraktora.

Az $\dot{x}=f(x,u)$ differenciálegyenlet-rendszert ($x\in R^n$, $u\in R^m$) *kvázigradiens-rendszernek* nevezzük, ha attraktorainak halmaza véges sok izoláltan elhelyezkedő egyensúlyi helyzetből áll, létezik $V(x,u)$ Ljapunov-függvénye, az utóbbinak (szigorú) minimumhelyei (rögzített u mellett x-ben) pontosan megegyeznek a rendszer attraktoraival és a rendszerre vonatkozó deriváltja csupán a rendszer egyensúlyi helyzeteiben zérus (egyébként negatív):

$$\dot{V}_{(3)}(x,u)=f(x,u)\cdot\operatorname{grad}_x V(x,u)\leqq 0. \tag{8}$$

Az elemi katasztrófaelmélet nyilvánvalóan érvényes kvázigradiens-rendszerekre is és ezek már az autonóm differenciálegyenlet rendszerek által generált dinamikai rendszereknek elegendően tág osztályát alkotják. Ez utóbbi megállapítás annak következménye, hogy LJAPUNOV direkt módszerének idézett tétele megfordítható, vagyis minden aszimptotikusan stabilis egyensúlyi helyzethez található alkalmas Ljapunov-függvény (lásd ZUBOV tételének megfordítását pl. [5]-ben). Természetesen elvileg is komoly gondot okozhat a különböző izolált attraktorokhoz tartozó Ljapunov-függvények megfelelő „illesztése".

Valamely rendszer Ljapunov-függvényének nem szükségképpen van fizikai (mechanikai, biológiai stb.) jelentése. Azonban gyakran előfordul, hogy „spekulatív úton konstruált" Ljapunov-függvényekről kiderül, hogy szemléletes jelentésük van, és fordítva, fontos fizikai tartalmat hordozó függvények gyakran kínálkoznak Ljapunov-függvényként. Ebből a szempontból is rendkívül érdekes és általános példát nyújt I. PRIGOGINE és P. GLANSDORFF elmélete, mely kísérletet tesz a termodinamikai egyensúlytól távoli, nyílt rendszerek stacionárius állapotainak, dinamikájá-

nak és stabilitásának jellemzésére [4], [7]. Ez az elmélet esetleg magyarázatot adhat az élettelen és az élő anyag fejlődésének ellentmondására, hogy ti. miért fejlődik az élet az egyre bonyolultabb, szervezettebb, strukturáltabb állapot felé akkor, amikor az élettelen rendszerek „fejlődése" ellenkező irányú. A Prigogine—Glansdorff-féle elmélet szerint, ha a peremfeltételek meggátolják a rendszert a termodinamikai egyensúly állapotának elérésében (melyben az entrópia állandó lenne), a rendszer az (időegység alatti) entrópia-növekedést igyekszik minimalizálni és a stacionárius állapot az entrópianövekedés minimumhelyén valósul meg. Ez a stacionárius állapot akkor stabilis, ha minden elég kicsi perturbációra az időegység alatti entrópianövekedés megváltozása pozitív definit. Az entrópianövekedés megváltozása az entrópia második differenciáljával van kapcsolatban, ami negatív definit. Ha tehát az entrópia második differenciáljának a rendszerre vonatkozó idő szerinti deriváltja pozitív definit, akkor az entrópia második differenciálja a rendszer Ljapunov-függvénye (−1-gyel meg kellene szorozni, hogy eredeti definíciónkkal összhangban legyünk), és ekkor a stacionárius állapot stabilis.

IRODALOM

[1] Барбашин, Е. А., *Введение в теорию устяочивости*, (Наука, Москва, 1967).
[2] Dodson, M. M., "Darwin's law of natural selection and Thom's theory of catastrophes", *Math. Biosc.* **28** (1976) 243—274.
[3] Farkas, M., „A szimultán tanulás dinamikai elmélete", *Alkalmazott Matematikai Lapok* **2** (1976) 103—114.
[4] Glansdorff, P. and Prigogine, I., *Thermodynamic theory of structure, stability and fluctuations* (Wiley—Interscience, London, New York, 1971).
[5] Hahn, W., *Stability of motion* (Springer, Berlin, Heidelberg, 1967).
[6] Peixoto, M., "Structural stability on two-dimensional manifolds", *Topology* **2** (1962) 101—121.
[7] Prigogine, I. and Glansdorff, P. «L'écart a l'équilibre interprété comme une source d'ordre. Structures dissipatives», *Bull. Cl. Sci. Acad. Roy. Belg. (5)* **59** (1973) 672—702.
[8] Сибирский, К. С., *Введение в топологическую динамику* (АНМССР, Кишинев, 1970).
[9] Thom, René, "Topological models in Biology", *Topology* **8** (1969) 313—335.
[10] Thom, René, *Stabilité structurelle et morphogénèse* (Benjamin, Reading, Mass., 1972).
[11] Trotman, D. and Zeeman, E. C., "The classification of elementary catastrophes of codimension ≦5", *Lecture Notes in Maths. 525* (Springer, 1975, Berlin, Heidelberg) 263—327.
[12] Zeeman, E. C., "Levels of structure in catastrophe theory illustrated by applications in the social and biological sciences", *Proc. ICM, 1974, Vancouver, vol. 2.* 533—546.

(Beérkezett: 1977. július 27.)

FARKAS MIKLÓS
BME GÉPÉSZMÉRNÖKI KAR MATEMATIKA TANSZÉK
1521 BUDAPEST XI., STOCZEK U. H ÉP. IV. EM.

ON QUALITATIVE CHARACTERIZATION OF PROCESSES

M. Farkas

Perhaps the vast majority of real (physical, biological, economic, social etc.) processes are *essentially deterministic* while their stochastic variations are only of secondary importance. Mathematical modelling of such processes developing in time can be accomplished with the help of *dynamical systems* provided that only the *attractors* and not all the individual points of phase space are considered to bear significance. *René Thom*'s elementary *catastrophe theory* is described and illustrated with some examples. The concept of a *quasi-gradient system* is introduced which extends the area of applicability of elementary catastrophe theory.

Alkalmazott Matematikai Lapok **2** *(1976) 259—268*

LINEÁRIS DIFFERENCIÁLEGYENLET-RENDSZER ILLESZTÉSE GRADIENS MÓDSZERREL

KANYÁR BÉLA ÉS TÓTH JÁNOS
Budapest

Megadjuk a gradiens módszer egy módosítását lineáris állandó együtthatós differenciálegyenlet-rendszerek illesztésére. A módszer az eltérések négyzetösszegét úgy minimalizálja, hogy a paraméterek szerinti parciális deriváltakat analitikusan számolja. Ezzel a differenciahányadossal való közelítés pontatlanságát és a paraméter-növekmény megválasztásának problémáját kiküszöböljük.
Megbecsültük a rendszer megoldásának és a parciális deriváltaknak a hibáját.
Az eljárás minden olyan módszer esetében alkalmazható, amely elsőrendű deriváltakat használ (pl. a *Gauss—Newton-módszer*, a *leggyorsabb leereszkedés* és a *Marquardt-eljárás*).

1. Bevezetés

A biológiai jelenségek kvantitatív leírásánál gyakran találkozunk a *paraméterbecslés* alábbi — speciális feladatával:
Adott egy

$$\frac{d\mathbf{y}}{dt} = \mathbf{A}\mathbf{y}, \quad \mathbf{y}(0) = \mathbf{y}_0 \tag{1.1}$$

kezdeti érték probléma, ahol az $\mathbf{A}=(a_{kl})$ mátrix valós, h-adrendű négyzetes mátrix, $\mathbf{y}$ a $[0, +\infty)$-ben értelmezett, értékeit $\mathbf{R}^h$-ban felvevő függvény, $\mathbf{y}_0 \in \mathbf{R}^h$ pedig tetszőleges, nemnegatív koordinátákkal bíró vektor. A kísérleti vizsgálatok értékelése úgy történik, hogy $\tilde{\mathbf{y}}(t_k)$ $(k=1, 2, \ldots, N)$ függvényértékeket mérünk, és ezekből akarunk következtetni az $\mathbf{A}$ mátrix elemeire. Pontosabban, azt az $\mathbf{A}$ mátrixot keressük, amely mellett a

$$\Phi\big(\mathbf{y}(t, \mathbf{A}), \tilde{\mathbf{y}}(t)\big) = \sum_{k=1}^{N} \big(\mathbf{y}(t_k, \mathbf{A}) - \tilde{\mathbf{y}}(t_k)\big)^T \big(\mathbf{y}(t_k, \mathbf{A}) - \tilde{\mathbf{y}}(t_k)\big) \tag{1.2}$$

hibafüggvény a minimumát veszi fel, vagyis a legkisebb négyzetek módszerét alkalmazzuk. (Φ alakját statisztikai megfontolások határozzák meg; sem ezekkel, sem más statisztikai kérdésekkel itt egyáltalán nem foglalkozunk.)
Helyezzük el az $\mathbf{A}$ mátrix elemeit egy $\mathbf{a}=(a_1, \ldots, a_{h^2})^T$ *paramétervektorban*. Ezzel a jelöléssel az (1.2)-ben szereplő Φ függvény minimalizálására használhatunk *gradiens módszereket* [3, 9], azaz eljárhatunk a következő módon:
Vegyünk egy tetszőleges $\mathbf{a}_0 \in \mathbf{R}^{h^2}$ pontot és ebből kiindulva képezzük az alábbi képlet segítségével $\mathbf{a}$ egymás utáni $\mathbf{a}_p$ közelítéseit:

$$\mathbf{a}_p = \mathbf{a}_{p-1} + \mathbf{S}^{-1}\nabla\Phi, \quad p = 1, 2, \ldots \tag{1.3}$$

ahol

$\nabla\Phi$: Φ-nek az **a** szerinti derivált vektora,

S: Φ-nek az **a** szerinti második derivált mátrixa.

Gyakran használt gradiens-eljárás a *Gauss—Newton-módszer* [3], ahol csak az elsőrendű deriváltakat kell képezni és a másodrendűeket az elsőrendűek segítségével közelítjük. Ez a módszer tehát a paramétervektor közelítését szintén az (1.3) képlet alapján számolja, ha **S** alatt most azt a mátrixot értjük, amelynek elemei:

$$s_{ij} = 2 \sum_{k=1}^{N} \left(\frac{\partial \mathbf{y}_k}{\partial a_i}\right)^T \left(\frac{\partial \mathbf{y}_k}{\partial a_j}\right), \quad i, j = 1, 2, \ldots, h^2, \tag{1.4}$$

ahol

$$\frac{\partial \mathbf{y}_k}{\partial a_i} = \left.\frac{\partial \mathbf{y}(t, \mathbf{A})}{\partial a_i}\right|_{t=t_k}, \quad N \geqq h^2 \text{ és feltesszük, hogy } \mathbf{S} \text{ invertálható.} \tag{1.5}$$

A *Gauss—Newton-módszer* segítségével tehát az (1.1) kezdeti érték problémában szereplő **A** mátrixnak megfelelő **a** paramétervektort úgy határozzuk meg, hogy kiszámítjuk

(i) egy tetszőleges $\mathbf{a}_0$ paramétervektor mellett (1.1) megoldását;
(ii) az (1.2) képletből Φ értékét az $\mathbf{a}_0$-nak megfelelő $\mathbf{A}_0$ helyen;
(iii) $\nabla\Phi$-t;
(iv) (1.4)-ből **S** elemeit;
(v) (1.3) felhasználásával $\mathbf{a}_1$-et.

Ezután elvégezzük az (i) lépést $\mathbf{a}_1$-gyel, s í.t. Alkalmas feltételek esetén $\mathbf{a}_p \to \mathbf{a}$ minden $\mathbf{a}_0$ esetén; ha $\mathbf{a}_p$ elég közel van **a**-hoz (vagy ha Φ értéke elég kicsiny), akkor megállunk ([1]).

Az (i) lépést lehetne *analitikusan* is végezni, hiszen (1.1) egy állandó együtthatós lineáris differenciálegyenlet-rendszerre vonatkozó kezdeti érték probléma. Ilyenkor **y** exponenciálisok összegeként fejezhető ki és Φ paraméterek szerinti deriváltjai is könnyen számolhatók. Azonban paraméterekként nem a közvetlen biológiai jelentéssel bíró mátrixelemek, hanem a mátrix sajátértékei és sajátvektorai szerepelnek. Igaz, hogy az utóbbiakból a mátrixelemek számolhatók, viszont a mátrixelemekre vonatkozó feltételek (pl. az, hogy közülük bizonyosak nulla értékűek, vagy pozitívak, némelyek viszonya adott) csak nehezen, vagy egyáltalán nem vehetők figyelembe az illesztés során. Ezért lényeges a differenciálegyenlet-rendszer közvetlen illesztése és az, hogy a mátrixelemeket tekinthessük paramétereknek.

Ha az (i) lépést *numerikusan* végezzük, akkor felmerül az a kérdés, hogy hogyan végezzük (iii)-at. BERMAN és munkatársai [2] numerikusan képezett derivált vektort használtak: a paraméterek minden egyes megváltoztatásánál megoldották a differenciálegyenlet-rendszert és az így kapható *differencia*hányadossal számoltak. Az eljárás nemlineáris rendszer esetén is alkalmazható, de pontatlan.

BUELL és munkatársai [4, 5] a derivált vektor meghatározását további differenciálegyenletek megoldására vezetik vissza, elkerülendő a paraméterek szerinti numerikus deriválást.

DICKINSON és GELINAS [6] — elsősorban paraméterérzékenységi vizsgálatok céljára — a derivált vektort szintén további differenciálegyenletek megoldásával kapják. Viszont megmutatják, hogy az újabb egyenletek megoldása visszavezethető

az eredeti differenciálegyenlet-rendszer *Jacobi mátrixa* inverzének képzésére. Módszerük a fenti feladatra specializálva egy h^2-rendű mátrix invertálását igényli.

Az ismertetendő módszer szintén nem differenciahányadossal közelíti a parciális deriváltakat, hanem az (1.1) kezdeti érték probléma megoldása közben analitikus formula alapján képezi ezeket. A gradiens módszerek közül a *Gauss—Newton-módszert* alkalmaztuk, tehát S-et (1.4)-ből számoltuk. A differenciálegyenlet-rendszer megoldását exponenciális sorfejtésből határoztuk meg ((i) lépés), ami állandó együtthatós esetben megegyezik a *Runge—Kutta-módszer*rel.

Az alábbiakban először megadjuk a parciális deriváltak számításához szükséges összefüggést, majd az előforduló lényeges mennyiségek hibájára felső becslést adunk. (A-val a pontos együttható-mátrixon kívül — amely tehát egy rögzített érték — Φ, illetve y változóját is jelöljük.)

2. A numerikus eljárás

Az eljárás az (i)—(v) alatt leírt lépésekből áll. A $\partial\Phi/\partial a_{kl}$ mennyiségekhez $\mathbf{y}(t, \mathbf{A})$ mellett a $\partial\mathbf{y}(t, \mathbf{A})/\partial a_{kl}$ parciális deriváltakat is ismerni kell. Ezeket a parciálisokat az (1.1) egyenlet

$$\mathbf{y}(t, \mathbf{A}) = \sum_{m=0}^{\infty} \left[\frac{t^m}{m!}\mathbf{A}^m\right]\mathbf{y}(0), \quad t \geqq 0 \tag{2.1}$$

megoldása felhasználásával a

$$\frac{\partial\mathbf{y}(t, \mathbf{A})}{\partial a_{kl}} = \left[\sum_{m=0}^{\infty}\frac{t^m}{m!}\mathbf{A}^m\right]\frac{\partial\mathbf{y}(0)}{\partial a_{kl}} + \left[\sum_{m=0}^{\infty}\frac{t^m}{m!}\frac{\partial\mathbf{A}^m}{\partial a_{kl}}\right]\mathbf{y}(0) \tag{2.2}$$

kifejezésből számolhatjuk. Ha az $\mathbf{y}(0)$ kezdőérték független a_{kl}-től, akkor (2.2) első tagja nulla, a második tag képzéséhez pedig a következő, könnyen verifikálható összefüggéseket lehet felhasználni:

$$\frac{\partial\mathbf{A}^m}{\partial a_{kl}} = \sum_{r=1}^{m}\mathbf{A}^{r-1}\frac{\partial\mathbf{A}}{\partial a_{kl}}\mathbf{A}^{m-r} = \sum_{r=1}^{m}\begin{bmatrix} a_{1k}^{(r-1)}a_{l1}^{(m-r)} & \dots & a_{1k}^{(r-1)}a_{lh}^{(m-r)} \\ a_{2k}^{(r-1)}a_{l1}^{(m-r)} & \dots & a_{2k}^{(r-1)}a_{lh}^{(m-r)} \\ \vdots & \ddots & \vdots \\ a_{hk}^{(r-1)}a_{l1}^{(m-r)} & \dots & a_{hk}^{(r-1)}a_{lh}^{(m-r)} \end{bmatrix}, \tag{2.3}$$

ahol $a_{ik}^{(r)}$ az $\mathbf{A}^r$ mátrix i-edik sorának k-adik eleme.

3. A véges sorfejtés hibájának becslése

Amennyiben a (2.1) kifejezésben az n-edik tagig megyünk el a sorfejtésben, a maradéktag:

$$\Delta\mathbf{y}_n(t) = \mathbf{y}(t) - \mathbf{y}_n(t) = \sum_{m=n+1}^{\infty}\frac{(\mathbf{A}t)^m}{m!}\mathbf{y}(0),$$

vagy az

$$\mathbf{R}_{n+1}(t) = \sum_{m=n+1}^{\infty}\frac{(\mathbf{A}t)^m}{m!}, \quad \mathbf{R}_{n+1}(t) = \left(r_{n+1;\,ij}(t)\right)$$

jelöléssel

$$\Delta \mathbf{y}_n(t) = \mathbf{R}_{n+1}(t)\mathbf{y}(0)$$

lesz, ahol $\mathbf{y}_n(t)$ jelöli $\mathbf{y}(t)$-nek azon közelítését, amelyet a (2.1) egyenlet jobboldalának első n tagját figyelembe véve kapunk. Az $\mathbf{R}_{n+1}(t)$ mátrix elemeire felső becslés adható.

3.1. TÉTEL. Ha bevezetjük az

$$\varepsilon_{n+2}(t) = \frac{\|\mathbf{A}\| t}{n+2}$$

mennyiséget, és feltesszük róla, hogy egynél kisebb, akkor ezt kapjuk:

$$(3.1) \qquad |r_{n+1;ij}(t)| \leqq \|\mathbf{R}_{n+1}(t)\| \leqq \sum_{m=n+1}^{\infty} \frac{(\|\mathbf{A}\| t)^m}{m!} \leqq \frac{(\|\mathbf{A}\| t)^{n+1}}{(n+1)!} \frac{1}{1-\varepsilon_{n+2}(t)}.$$

(Itt és a továbbiakban

$$\|\mathbf{X}\| = \max_k \sum_l |x_{kl}|, \quad \text{ha} \quad \mathbf{X} = (x_{kl}).)$$

A tétel *bizonyítása* triviális (KUO és KAISER [10]), a legutolsó egyenlőtlenség a végtelen mértani sor összegképletén alapul.

A továbbiak kedvéért vezessük be a következő jelölést:

$$\varrho(n) = \sum_{m=n+1}^{\infty} \frac{(\|\mathbf{A}\| t)^m}{m!}.$$

$\|\mathbf{A}\| t > 1$ esetén a (2.1) sorfejtés lassan konvergál és ekkor több (K) lépésben, $T = t/K$ lépésközzel érdemes integrálni a differenciálegyenlet-rendszert. Ezen eljárásra vonatkozó állításokat tartalmaz a

3.2. TÉTEL. Érvényesek az alábbi összefüggések:

a) $$\mathbf{y}(t) = \mathbf{y}(KT) = (\mathbf{R}_0(T))^K \mathbf{y}(0);$$

b) ha $\Delta \mathbf{y}(0) = \mathbf{0}$, akkor

$$\Delta \mathbf{y}_n(t) = -\sum_{j=0}^{K-1} \binom{K}{j} (\mathbf{R}_0(T))^j (-\mathbf{R}_{n+1}(T))^{K-j} \mathbf{y}(0);$$

c)

$$\|\Delta \mathbf{y}_n(t)\| \leqq \left[(\|\mathbf{R}_0(T)\| + \|\mathbf{R}_{n+1}(T)\|)^K - \|\mathbf{R}_0(T)\|^K\right] \|\mathbf{y}(0)\|;$$

végül

d) rögzített K és $n \to \infty$ esetén az $\varepsilon_n^*(T) = \|\mathbf{R}_{n+1}(T)\| / \|\mathbf{R}_0(T)\|$ jelöléssel

$$\|\Delta \mathbf{y}_n(t)\| \leqq K \cdot \varepsilon_n^*(T) \cdot \|\mathbf{R}_0(T)\|^K + O\left((\varepsilon_n^*(T))^2\right).$$

Bizonyítás: a) Teljes indukcióval belátható.

b) $\mathbf{y}_n(t)$ hibája ugyanis

$$\Delta \mathbf{y}_n(t) = (\mathbf{R}_0(T))^K \mathbf{y}(0) - (\mathbf{R}_0(T) - \mathbf{R}_{n+1}(T))^K \mathbf{y}(0).$$

c) Ha a b) állításban szereplő összeg normáját a tagok normájának összegével becsüljük, akkor ezt kapjuk:

$$\|\Delta \mathbf{y}_n(t)\| \leqq \sum_{j=0}^{K-1} \binom{K}{j} \|\mathbf{R}_0(T)\|^j \|\mathbf{R}_{n+1}(T)\|^{K-j} \|\mathbf{y}(0)\|,$$

ez pedig egyszerűbben éppen a kívánt módon írható.

d) A c)-ben szereplő jobboldalból $\|\mathbf{R}_0(T)\|^K$ kiemelhető; és mivel $n \to \infty$ esetén $\varepsilon_n^* \to 0$, éppen az állítást kapjuk.

Gyakorlati esetekben a 3.2. tétel a) állításában szereplő $(\mathbf{R}_0(T))^K$ mátrix elemeit (illetve ezeknek véges sorfejtéses közelítését) kell összehasonlítani a $K\varepsilon_n^*(T)\|\mathbf{R}_0(T)\|^K$ mennyiséggel és ez alapján növelni a sorfejtés tagjainak számát, vagy kisebb T lépésközt választani.

4. A parciális deriváltak hibájának becslése

A (2.2) kifejezésből következik, hogy a sorfejtésben az n-edik tagig elmenve a parciális deriváltak hibája:

$$\Delta \frac{\partial \mathbf{y}_n(t)}{\partial a_{kl}} = \frac{\partial \mathbf{y}(t)}{\partial a_{kl}} - \frac{\partial \mathbf{y}_n(t)}{\partial a_{kl}} = \sum_{m=n+1}^{\infty} \frac{t^m}{m!} \frac{\partial \mathbf{A}^m}{\partial a_{kl}} \mathbf{y}(0), \tag{4.1}$$

vagy az

$$\mathbf{U}_{n+1;kl}(t) = \sum_{m=n+1}^{\infty} \frac{t^m}{m!} \frac{\partial \mathbf{A}^m}{\partial a_{kl}}$$

jelöléssel

$$\Delta \frac{\partial \mathbf{y}_n(t)}{\partial a_{kl}} = \mathbf{U}_{n+1;kl}(t)\mathbf{y}(0) \tag{4.2}$$

lesz. (Az $\mathbf{U}_{n+1;kl}(t)$ mátrixok száma egyenlő az $\mathbf{A}$ mátrix elemeinek a számával, h^2-tel.) Az $\mathbf{U}_{n+1;kl}(t)$ mátrix elemeire felső becslés adható.

4.1. TÉTEL: Elhagyva az $\mathbf{U}_{n+1;kl}(t)$ mátrix $u_{n+1;kl;ij}(t)$ elemének és $\mathbf{A}$ elemeinek az indexeit és az első változóját és feltéve, hogy $\varepsilon_{n+1}(t) < 1$, ezt kapjuk:

$$|\mathbf{u}| \leqq \|\mathbf{U}\| \leqq t \cdot \sum_{m=n}^{\infty} \frac{(\|\mathbf{A}\| t)^m}{m!} \leqq \frac{(\|\mathbf{A}\| \cdot t)^n t}{n!} \frac{1}{1-\varepsilon_{n+1}(t)}. \tag{4.3}$$

Bizonyítás: Csak a második egyenlőtlenséget bizonyítjuk:

$$\|\mathbf{U}\| \leqq \sum_{m=n+1}^{\infty} \frac{t^m}{m!} \left\| \frac{\partial \mathbf{A}^m}{\partial a} \right\| \leqq \sum_{m=n+1}^{\infty} \frac{t^m}{m!} \sum_{r=1}^{m} \|\mathbf{A}\|^{r-1} \left\| \frac{\partial \mathbf{A}}{\partial a} \right\| \|\mathbf{A}\|^{m-r} =$$

$$= t \cdot \sum_{m=n}^{\infty} \frac{(\|\mathbf{A}\| \cdot t)^m}{m!} = t\varrho(n-1).$$

Az első egyenlőtlenséget (2.3) felhasználásával kaptuk, az utolsó egyenlőséget a továbbiak kedvéért írtuk fel.

A függvény hibabecslésénél kapott $\varrho(n)$ és a parciális deriváltaknál kapott $t\varrho(n-1)$ becslések összehasonlításával belátható, hogy n növelése az elsőt bizonyos értelemben jobban csökkenti, mint a másodikat.

4.2. TÉTEL: A fenti jelölésekkel $t>0$ és $\|\mathbf{A}\|t<n+3$ esetén fennáll, hogy

$$\frac{\varrho(n+1)}{\varrho(n)} < \frac{t\varrho(n)}{t\varrho(n-1)}, \tag{4.4}$$

ha pedig $\|\mathbf{A}\|<n+1$, akkor

$$\varrho(n+1)-\varrho(n) > t\varrho(n)-t\varrho(n-1). \tag{4.5}$$

Bizonyítás: (4.4) így is írható:

$$\frac{\varrho(n+1)}{\varrho(n)} < \frac{\varrho(n)}{\varrho(n-1)} = \frac{(\|\mathbf{A}\|t)^{n+1}/(n+1)!+\varrho(n+1)}{(\|\mathbf{A}\|t)^{n}/n!+\varrho(n)}. \tag{4.6}$$

Átrendezés után látható, hogy (4.6) teljesüléséhez elegendő

$$\frac{\varrho(n+1)}{\varrho(n)} < \frac{\|\mathbf{A}\|t}{n+1} \tag{4.7}$$

vagy az ezzel ekvivalens

$$\varrho(n+1) < \frac{(\|\mathbf{A}\|t)^{n+1}}{(n+1)!}\,\frac{1}{(n+1)/\|\mathbf{A}\|t-1} \tag{4.8}$$

egyenlőtlenség teljesülése. (3.1)-ből viszont következik, hogy

$$\varrho(n+1) < \frac{(\|\mathbf{A}\|t)^{n+2}}{(n+2)!}\,\frac{1}{1-\|\mathbf{A}\|t/(n+3)}. \tag{4.9}$$

(4.9) jobb oldala megegyezik az alábbi szorzattal:

$$\left(\frac{(\|\mathbf{A}\|t)^{n+1}}{(n+1)!}\,\frac{1}{(n+1)/\|\mathbf{A}\|t-1}\right)\left(\frac{\|\mathbf{A}\|t}{n+2}\,\frac{(n+1)/\|\mathbf{A}\|t-1}{1-\|\mathbf{A}\|t/(n+3)}\right) \tag{4.10}$$

s ezen szorzat második tényezője egynél kisebb, így (4.8), s ezzel együtt a bizonyítandó (4.4) egyenlőtlenség is teljesül.

(4.5) így is írható:

$$-\frac{(\|\mathbf{A}\|t)^{n+1}}{(n+1)!} < -t\cdot\frac{(\|\mathbf{A}\|t)^{n}}{n!}, \tag{4.11}$$

(4.11) pedig egyenértékű a tételben tett feltevéssel (ami gyakorlati esetekben általában teljesül).

Többlépéses, T állandó lépésközű integrálás esetén is megbecsülhetjük a parciális deriváltak hibáját. Ekkor a hibát megadja a

4.3. TÉTEL.

$$\Delta\frac{\partial \mathbf{y}_n(t)}{\partial a_{kl}} = \mathbf{D}_{n;kl}(T)\,\mathbf{y}(0), \tag{4.12}$$

ahol

$$\mathbf{D}_{n;kl}(T) = \sum_{j=0}^{K-1} (\mathbf{R}_0(T))^j \{2\mathbf{U}_{0;kl}(T) - [\mathbf{I} - \mathbf{R}_{n+1}(T)/\mathbf{R}_0(T)]^j \mathbf{U}_{0;kl}(T) - \\ - \mathbf{U}_{0;kl}(T)[\mathbf{I} - \mathbf{R}_{n+1}(T)/\mathbf{R}_0(T)]^{K-j-1} + \mathbf{U}_{n+1;kl}(T)\}(\mathbf{R}_0(T))^{K-j-1}, \tag{4.13}$$

itt $\mathbf{I}$ az egységmátrix.

Bizonyítás: Nyilvánvalóan fennáll a (2.2)-höz hasonló

$$\frac{\partial \mathbf{y}(KT)}{\partial a_{kl}} = \mathbf{R}_0(T) \frac{\partial \mathbf{y}((K-1)T)}{\partial a_{kl}} + \mathbf{U}_{0;kl}(T)(\mathbf{R}_0(T))^{K-1}\mathbf{y}(0) \tag{4.14}$$

rekurzív formula a parciális deriváltakra. A (4.14) rekurzív formula kifejtéséből a

$$\frac{\partial \mathbf{y}(KT)}{\partial a_{kl}} = \left[\sum_{j=0}^{K-1} (\mathbf{R}_0(T))^j \mathbf{U}_{0;kl}(T)(\mathbf{R}_0(T))^{K-j-1}\right] \mathbf{y}(0) \tag{4.15}$$

összefüggést kapjuk. Így a parciális deriváltak hibája:

$$\Delta \frac{\partial \mathbf{y}_n(t)}{\partial a_{kl}} = \frac{\partial \mathbf{y}(t)}{\partial a_{kl}} - \frac{\partial \mathbf{y}_n(t)}{\partial a_{kl}} = \left[\sum_{j=0}^{K-1} \Delta(\mathbf{R}_0(T))^j \mathbf{U}_{0;kl}(T)(\mathbf{R}_0(T))^{K-j-1} + \right. \tag{4.16}$$

$$\left. + \sum_{j=0}^{K-1} (\mathbf{R}_0(T))^j \Delta \mathbf{U}_{0;kl}(T)(\mathbf{R}_0(T))^{K-j-1} + \sum_{j=0}^{K-1} (\mathbf{R}_0(T))^j \mathbf{U}_{0;kl}(T) \Delta(\mathbf{R}_0(T))^{K-j-1}\right] \mathbf{y}(0) =$$

$$= \sum_{j=0}^{K-1} \{[(\mathbf{R}_0(T))^j - (\mathbf{R}_0(T) - \mathbf{R}_{n+1}(T))^j] \mathbf{U}_{0;kl}(T)(\mathbf{R}_0(T))^{K-j-1} +$$

$$+ (\mathbf{R}_0(T))^j \mathbf{U}_{n+1;kl}(T)(\mathbf{R}_0(T))^{K-j-1} +$$

$$+ (\mathbf{R}_0(T))^j \mathbf{U}_{0;kl}(T)[(\mathbf{R}_0(T))^{K-j-1} - (\mathbf{R}_0(T) - \mathbf{R}_{n+1}(T))^{K-j-1}]\} \mathbf{y}(0).$$

A (4.16) egyenlőségsorozat legutolsó összegének minden tagjából balról $(\mathbf{R}_0(T))^j$-t, jobbról $(\mathbf{R}_0(T))^{K-j-1}$-et kiemelve a bizonyítandó (4.13) egyenlőséget kapjuk.

A hiba becsléséről szól az utolsó, a

4.4. TÉTEL: Rögzített K és $n \to \infty$ esetén fennáll, hogy

$$|d_{ij}| \leqq \|\mathbf{D}\| \leqq KTe^{(K-1)\|\mathbf{A}\|T}\left[(K-1)\varepsilon_n^*(T)e^{\|\mathbf{A}\|T} + \frac{(\|\mathbf{A}\|T)^n}{n!}\frac{1}{1-\varepsilon_{n+1}(T)}\right] + O((\varepsilon_n^*(T))^2). \tag{4.17}$$

(Itt és az alábbiakban a mátrixok legtöbb indexét és argumentumát nem írjuk ki, $\mathbf{D}$ elemeit d_{ij}-vel jelöljük.)

Bizonyítás: (4.13)-ból kiindulva kapjuk:

$$|d_{ij}| \leqq \|\mathbf{D}\| \leqq \sum_{j=0}^{K-1} \|\mathbf{R}_0(T)\|^{K-1}\left[(K-1)\varepsilon_n^*(T)\|\mathbf{U}_0\| + \|\mathbf{U}_{n+1}\| + o(\varepsilon_n^*(T))\right] =$$

$$= K\|\mathbf{R}_0(T)\|^{K-1}\left[(K-1)\varepsilon_n^*(T)\|\mathbf{U}_0\| + \|\mathbf{U}_{n+1}\|\right] + o(\varepsilon_n^*(T)) \leqq \tag{4.18}$$

$$\leqq KTe^{(K-1)\|\mathbf{A}\|T}\left[(K-1)\varepsilon_n^*(T)\,e^{\|\mathbf{A}\|T} + \frac{(\|\mathbf{A}\|T)^n}{n!}\frac{1}{1-\varepsilon_{n+1}(T)}\right] + o(\varepsilon_n^*(T))$$

ahol az utolsó átalakításnál felhasználtuk az

$$\|\mathbf{R}_0(T)\| \leqq \sum_{m=0}^{\infty} \frac{(\|\mathbf{A}\|T)^m}{m!} = e^{\|\mathbf{A}\|T}$$

és

$$\|\mathbf{U}_0(T)\| \leqq T \cdot \sum_{m=0}^{\infty} \frac{(\|\mathbf{A}\|T)^m}{m!} = Te^{\|\mathbf{A}\|T}$$

nyilvánvaló egyenlőtlenségeket, valamint a 4.1. tétel állítását $t=T$ esetére.

Így K és T ismeretében $\|\mathbf{D}\|$ is becsülhető, majd a (4.15)-ben $\mathbf{y}(0)$ előtt álló együttható-mátrix minden elemével összehasonlítható. Amennyiben a $\|\mathbf{D}\|$ becsült értéke a mátrixelemekhez viszonyítva nagy, az n értékét növelni, vagy a T értékét csökkenteni kell.

Felhívjuk a figyelmet arra, hogy a becslésekben az ismeretlen $\|\mathbf{A}\|$ szerepel, gyakorlati esetekben tehát csak közelítő becsléseket kaphatunk $\mathbf{A}$ számolt elemeinek felhasználásával.

5. Numerikus példa

A megadott eljárást először három differenciálegyenlettel és négy paraméterrel leírható rekeszmodellnél próbáltuk ki IBM 360/75 számítógépen, FORTRAN nyelvű program segítségével [8]. Nemlineáris regresszióhoz felhasználtuk az UCLA-ban készült BMDX 85 programot [7], amely Gauss—Newton—Hartley-féle módszeren alapul. Programunk továbbfejlesztett változatban R—20-as gépen készült el. Tesztelésre — többek között — egy négy differenciálegyenlet-rendszerrel és 5 paraméterrel leírható modellt választottunk. Az öt paraméter között egy, a mért érték és az egyenletrendszer megoldása közötti szorzótényező volt, azaz lineáris paraméterként szerepelt. A paraméterek a differenciálegyenlet-rendszer mátrixát az

$$\mathbf{A} = \begin{pmatrix} -p_2 & p_1 & 0 & 0 \\ p_2 & -(p_1+p_3) & 0 & 0 \\ 0 & p_3 & -p_4 & 0 \\ 0 & 0 & p_4 & 0 \end{pmatrix}$$

alakban határozták meg. Mérési adatokat az 1. táblázatban található valódi paraméterértékekkel és $y_1(0)=100$, $y_2(0)=y_3(0)=y_4(0)=0$ kezdőfeltételekkel a *Runge—Kutta-módszer*rel [9] generáltunk. Mivel szimultán két függvény illesztését kívántuk kipróbálni az adatokat $f_1=y_1$ és $f_2=p_5\,y_3$ szerint számoltuk 2×12 pontban az

$1 \leqq t \leqq 70$ intervallumban. A pontosan számolt y_1 és y_3 értékeket a 4. jegy elhagyásával kerekítettük.

Több különböző iterációs kezdő paraméterértékek mellett szimultán illesztve az f_1 és f_2 függvényeket, mindig egy minimumot kaptunk, az 1. táblázatban található becsült paraméter- és szórásértékekkel. Ezek az értékek jó egyezést mutatnak a valódi értékekkel

1. TÁBLÁZAT

A valódi és a becsült paraméterértékek a becsült szórásokkal

	p_1	p_2	p_3	p_4	p_5
valódi érték	0,025000	0,100000	0,0080000	0,080000	1,00000
becsült érték	0,024846	0,099975	0,0079841	0,080001	1,00025
becsült szórás	±0,000026	±0,000076	±0,0000128	±0,000134	±0,00034

Tapasztalatunk szerint a hibabecsléshez használt, 3.2. tétel c) összefüggésének hátránya — a könnyű számolhatóság előnye mellett —, hogy túl szigorú. Példánkban hibakorlátként 0,2 és 10 értékeket választva, mind a paraméterértékek, mind a számolt y megoldások eltérése kisebb mint 1% maradt. A 10-es hibakorlát mellett a felhasznált gépidő felére csökkent.

Egy két paraméterrel és két egyenlettel leírható modell esetén 2—3 perc, az említett példa (4 egyenlet és 5 paraméter) esetén pedig 7—10 perc szükséges R—20 számítógépen. Természetesen az időszükséglet erősen függ a mérési pontok számától és a paraméter kezdőértékektől is.

Köszönettel tartozunk a lektoroknak részletes és alapos bírálatukért.

IRODALOM

[1] BARD, Y., *Nonlinear Parameter Estimation* (Acad. Press. New York, London, 1974).
[2] BERMAN, M. and M. F. WEISS, SAAM (Simulation, Analysis and Modelling) Manual. Bethesda 1966.
[3] BOX, M. J., D. DAVIES and W. H. SWANN, *Nonlinear Optimization Techniques*. (Oliver and Boyd, Edinburgh, 1969).
[4] BUELL, J., R. KALABA and E. RUSPINI, "Identification of linear systems using long periods of observation", *J. Optimization Theory Appl.* **5** (1970) 170—177.
[5] BUELL, J., R. KALABA, A. YAKUSH and E. RUSPINI, "A Program for Identification of Linear Systems", *Comput. Progr. Biomed. 2*, 8—15 (1971).
[6] DICKINSON, R. P. and R. J. GELINAS, "Sensitivity analysis of ordinary differential equation systems — A direct method", *J. Comp. Phys.* **21** (1976) 123—143.
[7] DIXON, W. J. (ed.), *Biomedical Computer Programs* (Univ. Calif. Press, Los Angeles, 1972).
[8] KANYÁR, B., „Anyagcseremérések kiértékelése rekesz modellek segítségével. II. Modellek és mérési eredmények illesztése", *Orvos és Technika* **11** (1975) 161—168.
[9] KIS, O., *Számítási módszerek I., II. (jegyzet)* (Tankönyvkiadó, Budapest, 1969).
[10] KUO, F. F. and J. F. KAISER, *System Analysis by Digital Computer* (Wiley, New York, London, 1966).
[11] MARQUARDT, D. W., "An algorithm for least-squares estimation of nonlinear parameters", *SIAM J.* **11** (1963) 431—441.

(Beérkezett: 1976. január 13.)

(Újra beérkezett: 1976. december 6.)

KANYÁR BÉLA ÉS TÓTH JÁNOS
SEMMELWEIS OTE SZÁMÍTÁSTECHNIKAI CSOPORT
1089 BUDAPEST, VIII., KULICH GY. TÉR 5.

FITTING OF A SYSTEM OF LINEAR DIFFERENTIAL EQUATIONS BY GRADIENT METHOD

B. KANYÁR and J. TÓTH

For fitting of a system of linear differential equations with constant coefficients a gradient method is given. It minimizes the sum of squares of residuals by calculating the partial derivatives with respect to the parameters by analytical differentiation. Therefore the inaccuracy of numerical differentiation and the problem of choosing the increment in the parameters are eliminated.

The error of the solution of the system and that of the partial derivatives as well are estimated.

The procedure proposed can be used in the case of all of the methods with first order partial derivatives (e.g. *Gauss—Newton, steepest descent* and *Marquardt-methods*).

Alkalmazott Matematikai Lapok **2** *(1976) 269—276*

PERIODIKUS DIFFERENCIÁLEGYENLET-RENDSZEREK KARAKTERISZTIKUS MULTIPLIKÁTORAINAK KÖZELÍTŐ MEGHATÁROZÁSÁRÓL

KOTSIS DOMOKOSNÉ
Budapest

Ez a dolgozat periodikus együtthatójú lineáris differenciálegyenlet-rendszerek karakterisztikus multiplikátorainak közelítő meghatározásával foglalkozik. Az eljárás DEMIDOVICS egy tételén alapul. Algebrai feltételeket ismertetünk a karakterisztikus multiplikátorok elhelyezkedésére, majd a közelítő karakterisztikus multiplikátor hibabecslését ismertetjük. Végül két fontos speciális esetben szólunk a karakterisztikus multiplikátorok konkrét kiszámításáról.

1. Bevezetés

Nem lineáris autonóm, vagy periodikus differenciálegyenlet-rendszer periodikus megoldásának stabilitása a variációs rendszer segítségével vizsgálható. A variációs rendszer egy periodikus együtthatójú lineáris differenciálegyenlet-rendszer, mely főmátrixának sajátértékei a karakterisztikus multiplikátorok. A karakterisztikus multiplikátorok komplex számsíkon való elhelyezkedése alapján következtethetünk a periodikus megoldások stabilitására, aszimptotikus stabilitására, s nem utolsó sorban a periodikus megoldások izoláltságát is ezen multiplikátorok elhelyezkedése alapján lehet vizsgálni. ([5], [2].)

A vizsgálatok jelentős részében azt kell eldöntenünk, hogy a karakterisztikus multiplikátorok a komplex számsík $|z|<1$ tartományában helyezkednek-e el. A komplex számsíkon bevezetett $z=\dfrac{\eta+1}{\eta-1}$ lineáris törtleképezés ezt a körtartományt a $\operatorname{Re}\eta<0$ tartományba képezi le, s ilyen módon a karakterisztikus multiplikátorok, melyek egy

$$f_n(z) = a_n z^n + \ldots + a_1 z + a_0 \tag{1.1}$$

polinom gyökei, akkor és csak akkor helyezkednek el az origó körüli egységsugarú kör belsejében, ha az

$$(\eta-1)^n \cdot f_n\left(\frac{\eta+1}{\eta-1}\right) = a_n(\eta+1)^n + \ldots + a_0(\eta-1)^n \tag{1.2}$$

polinom stabilis. Másod- és harmadfokú polinomok esetén, $a_n=1$ választással ennek feltétele:

$$n=2:\ |a_1| < 1+a_0,\quad a_0 < 1; \tag{1.3}$$

$$n=3:\ 1-a_2+a_1-a_0 \gtrless 0,\quad 3-a_2-a_1+3a_0 \gtrless 0,\quad 1+a_2+a_1+a_0 \gtrless 0; \tag{1.4a}$$

$$1-a_1-a_0^2+a_0a_2 > 0. \tag{1.4b}$$

(1.4a) mindhárom egyenlőtlenségében egyidejűleg vagy a $>$, vagy a $<$ relációnak kell fennállni (l. [3]).

A karakterisztikus multiplikátorok kiszámítására közelítő eljárás alkalmazható (l. [1]).

Legyen adott az

$$\dot{x} = A(t)x \tag{1.5}$$

differenciálegyenlet-rendszer, ahol x n-dimenziós, $A(t)$ $n\times n$-es mátrixfüggvény, elemei valós változós (általában komplex értékű) függvények, $a_{ik}\in C_R^0$, $A(t)=[a_{ik}(t)]$, $a_{ik}(t+\tau)\equiv a_{ik}(t)$, $i, k=1, 2, \ldots, n$, $\tau>0$ periódus. Jelölje $X(t)$ (1.5) azon alapmátrixát, melyre $X(0)=E$ (egységmátrix). Jelölje továbbá $X_h(\tau)$ az $X(\tau)$ főmátrix közelítését:

$$X_h(\tau) = e^{hA_{m-1}}e^{hA_{m-2}}\ldots e^{hA_1}e^{hA_0}, \tag{1.6}$$

ahol $h=\frac{\tau}{m}$ ($m\geqq 1$ egész), A_k pedig $A(t)$-nek közelítése a $[kh, (k+1)h)$ intervallumban a következőképpen:

$$\min_{t\in[kh,(k+1)h]} A(t) \leqq A_k \leqq \max_{t\in[kh,(k+1)h]} A(t) \tag{1.7}$$

(mátrix minimumát, ill. maximumát elemenként képezve). A B mátrixnak két normáját használjuk:

$$\|B\|_1 = \max_j \sum_k |b_{jk}|, \tag{1.8}$$

$$\|B\|_2 = n \max_{j,k} |b_{jk}|. \tag{1.9}$$

1.1. Tétel: Minden $\varepsilon>0$-hoz van olyan $m\geqq 1$, hogy ha $\delta(\varepsilon)<\frac{\tau}{m}=h$, akkor

$$\|X(\tau)-X_h(\tau)\|_1 < \varepsilon\tau e^{M(\tau+h)}, \tag{1.10}$$

ahol $\delta(\varepsilon)$ az $A(t)$ mátrixfüggvény folytonossági modulusa ($\|A(t')-A(t'')\|_1<\varepsilon$, ha $|t'-t''|<\delta(\varepsilon)$), továbbá $\|A(t)\|_1\leqq M$. (L. [1].)

Ezen tétel felhasználásával a főmátrix sajátértékei előírt pontossággal közelítő mátrix sajátértékeiként származtathatók.

2. A karakterisztikus multiplikátorok hibájának becslése

Másod- és harmadfokú karakterisztikus polinomok esetén pontosabban meghatározható a sajátértékek hibája, ha (1.10) alapján közelítő mátrixból indulunk ki. Vizsgáljuk először a 2-dimenziós esetet. Legyenek a karakterisztikus polinom együtthatói

$$c_2 = 1, c_1, c_0,$$

s a közelítő karakterisztikus polinom együtthatói

$$c_2^h = 1, c_1^h, c_0^h.$$

Könnyen látható, hogy

$$|c_1 - c_1^h| \leqq \|X(\tau) - X_h(\tau)\|_2 < 2\varepsilon\tau e^{M(\tau+h)}, \tag{2.1}$$

$$|c_0 - c_0^h| \leqq \sup \|X(t)\|_2 \|X(\tau) - X_h(\tau)\|_2 < 2\varepsilon\tau e^{M(2\tau+h)}. \tag{2.2}$$

Hasonlóan 3-dimenziós esetben is becsülhetők az együtthatók eltérései; itt

$$c_3 = 1, c_2, c_1, c_0, \quad \text{ill.} \quad c_3^h = 1, c_2^h, c_1^h, c_0^h$$

a megfelelő pontos, ill. közelítő együtthatók, ezekre a következő egyenlőtlenségek érvényesek:

$$|c_2 - c_2^h| \leqq \|X(\tau) - X_h(\tau)\|_2 < 3\varepsilon\tau e^{M(\tau+h)}, \tag{2.3}$$

$$|c_1 - c_1^h| \leqq \frac{4}{3} \sup \|X(t)\|_2 \|X(\tau) - X_h(\tau)\|_2 < 4\varepsilon\tau e^{M(2\tau+h)}, \tag{2.4}$$

$$|c_0 - c_0^h| \leqq \frac{2}{3} \sup \|X(t)\|_2^2 \|X(\tau) - X_h(\tau)\|_2 < 2\varepsilon\tau e^{M(3\tau+h)}. \tag{2.5}$$

Jelölje a továbbiakban $\delta(c_1)$, $\delta(c_0)$, ill. $\delta(c_2)$, $\delta(c_1)$, $\delta(c_0)$ a c_1, c_0, ill. a c_2, c_1, c_0 együtthatók (2.1), (2.2), ill. (2.3), (2.4), (2.5)-beli hibakorlátját. Vizsgáljuk meg, hogy ezen hibák hogy befolyásolják a sajátértékeket. Kétdimenziós esetben — felhasználva a gyökök egzakt meghatározását — a ϱ tényleges és ϱ^h közelítő gyök eltérése a következőképpen becsülhető:

$$|\varrho - \varrho^h| \leqq \frac{\delta(c_1)}{2} + \frac{1}{2} \left| \sqrt{c_1^h - 4c_0^h} - \sqrt{c_1 - 4c_0} \right|, \tag{2.6}$$

amiből, felhasználva a $\left|\sqrt{P+Q} - \sqrt{P}\right| \leqq \sqrt{|Q|}$, $P, Q \in R$ azonos egyenlőtlenséget, a

$$|\varrho - \varrho^h| \leqq \delta(c_1) + \sqrt{\frac{\delta(c_1)\,|c_1^h|}{2}} + \sqrt{\delta(c_0)} \tag{2.7}$$

egyenlőtlenség adódik (itt $\delta(c_1)$ és $\delta(c_0)$ (2.1) és (2.2)-vel helyettesíthető, $|c_1^h| = |\mathrm{Sp}\, X^h(\tau)| \leqq e^{M\tau}$).

A fenti hibabecslés másodfokú egyenlet egyszeres gyökeinél helyettesíthető egy közelítő hibabecsléssel (l. [6]). Ha $\varrho^h \neq -\frac{c_1}{2}$, akkor a másodfokú polinomot a ϱ^h helyhez tartozó elsőfokú *Taylor-polinom*jával helyettesítve a

$$|\varrho - \varrho^h| \approx \left| \frac{c_0^h - c_0 + \varrho^h(c_1^h - c_1)}{2\varrho^h + c_1} \right| \tag{2.8}$$

közelítő egyenlőség adódik a gyökök hibájára. Egyébként a másodfokú *Taylor-polinom*ot alkalmazva a

$$|\varrho - \varrho^h| = \left| \sqrt{\varrho^h(c_1 - c_1^h) + (c_0 - c_0^h)} \right|$$

egyenlőség, s ebből a

$$|\varrho-\varrho^h| \leqq \sqrt{|\varrho^h|\delta(c_1)+\delta(c_0)} \tag{2.9}$$

becslés következik.

Háromdimenziós esetben a polinomok egzakt gyökeinek előállítására egyszerű módszer nem használható. Itt is a fenti közelítő hibabecslés alkalmazható:

$$|\varrho-\varrho^h| \approx \left|\frac{\varrho^{h2}(c_2-c_2^h)+\varrho^h(c_1-c_1^h)+c_0-c_0^h}{3\varrho^{h2}+2c_2\varrho^h+c_1}\right|. \tag{2.10}$$

Autonóm differenciálegyenlet-rendszer (nem triviális) periodikus megoldására, ill. vektorváltozójában periodikus autónom differenciálegyenlet-rendszer ún. D-periodikus megoldására vonatkozó variációs rendszer karakterisztikus multiplikátorai között az 1 szám előfordul. Nyilvánvaló, hogy annak feltételezésével, hogy az egyik gyök 1, a kétdimenziós eset 1-dimenziósra, a 3-dimenziós 2-dimenziósra redukálódik. Kétdimenziós esetben a feltételi egyenlet:

$$1+c_1+c_0 = 0, \tag{2.11}$$

ami azt jelenti, hogy az egyik gyök éppen c_0-lal egyenlő. Ekkor

$$|\varrho-\varrho^h| \leqq \delta(c_0), \tag{2.12}$$

ha a gyök számításánál c_0^h értékét használjuk, ill.

$$|\varrho-\varrho^h| \leqq \delta(-1-c_1) = \delta(c_1), \tag{2.13}$$

ha a számításnál c_1^h-t használjuk.

Hasonlóan háromdimenziós esetben annak feltétele, hogy az 1 szám gyök legyen az

$$1+c_2+c_1+c_0 = 0 \tag{2.14}$$

egyenlőség fennállása. Ez azt jelenti, hogy (2.7) felhasználásával a gyökök hibakorlátjára a

$$\delta_{10} = \delta(c_0)+\delta(c_1)+\sqrt{\frac{\delta(c_0)+\delta(c_1)}{2}|c_0+c_1|}+\sqrt{\delta(c_0)}, \tag{2.15}$$

$$\delta_{20} = \delta(c_2)+\sqrt{\frac{\delta(c_2)}{2}|c_2+1|}+\sqrt{\delta(c_0)}, \tag{2.16}$$

$$\delta_{21} = \delta(c_2)+\sqrt{\frac{\delta(c_2)}{2}|c_2+1|}+\sqrt{\delta(c_2)+\delta(c_1)} \tag{2.17}$$

értékek adódnak annak megfelelően, hogy a gyökök számításánál melyik két együtthatót használtuk.

Ezen esetekben megfogalmazhatók azok a feltételek, melyek elégségesek ahhoz, hogy a gyökök — az 1 kivételével — az origó középpontú egységsugarú kör belsejében helyezkedjenek el.

Kétdimenziós egyenlet esetén az (1.3) feltétel (2.11) felhasználásával a

$$|c_0| < 1, \tag{2.18}$$

ill. számított c_0^h esetén a

$$|c_0^h| < 1-\delta(c_0) \tag{2.18'}$$

egyenlőtlenségre redukálódik.

Háromdimenziós esetben (1.4)-et az alábbi egyenlőtlenségek egyike helyettesíti aszerint, hogy melyik két együtthatóval számítjuk a gyököket:

$$\begin{cases} 3+2c_2+c_1 > 0 \\ c_2+c_1 < 0 \\ c_1+1 > 0, \end{cases} \tag{2.19}$$

$$\begin{cases} 2+c_2-c_0 > 0 \\ c_0+1 > 0 \\ c_2+c_0 < 0, \end{cases} \tag{2.20}$$

$$\begin{cases} 1-2c_0-c_1 > 0 \\ c_0+1 > 0 \\ c_1+1 > 0. \end{cases} \tag{2.21}$$

Közelítő pontosságú együtthatók esetén (2.19)-et a

$$\begin{cases} 3+2c_2^h+c_1^h > \delta(c_1)+2\delta(c_2) \\ c_2^h+c_1^h+\delta(c_1)+\delta(c_2) < 0 \\ c_1^h+1 > \delta(c_1), \end{cases} \tag{2.19'}$$

(2.20)-at a

$$\begin{cases} 2+c_2^h-c_0^h > \delta(c_0)+\delta(c_2) \\ c_0^h+1 > \delta(c_0) \\ c_2^h+c_0^h+\delta(c_0)+\delta(c_2) < 0, \end{cases} \tag{2.20'}$$

(2.21)-et pedig az

$$\begin{cases} 1-2c_0^h-c_1^h > \delta(c_1)+2\delta(c_0) \\ c_0^h+1 > \delta(c_0) \\ c_1^h+1 > \delta(c_1) \end{cases} \tag{2.21'}$$

egyenlőtlenségrendszer helyettesíti.

3. Speciális másod- és harmadrendű differenciálegyenletek karakterisztikus multiplikátorainak kiszámítása

Tekintsük először az

$$\ddot{x}+p(x^2-1)\dot{x}+x = 0, \quad p > 0 \tag{3.1}$$

ún. *Van der Pol-differenciálegyenletet.* Tudjuk, hogy (3.1)-nek létezik $u_0(t)$ periodikus megoldása $\tau_0>0$ periódussal. Ezen (nem konstans) periodikus megoldásra

vonatkozó első variációs rendszer az

$$\dot{y} = \begin{bmatrix} 0 & 1 \\ -1 & -p(u_0^2(t)-1) \end{bmatrix} y \tag{3.2}$$

lineáris differenciálegyenlet-rendszer (y 2-dimenziós vektor). A főmátrix számítható (1.6) segítségével, ahol

$$A_k = \begin{bmatrix} 0 & 1 \\ -1 & a_k \end{bmatrix}, \quad a_k = -p(u_0^2(t_k)-1), \quad t_k = k\frac{\tau_0}{m}, \quad k = 0, 1, \ldots, m-1. \tag{3.3}$$

Mint könnyen kiszámítható, $|a_k| \neq 2$ esetén

$$e^{hA_k} = \frac{e^{h\frac{a_k}{2}}}{b_k} \begin{bmatrix} -\frac{a_k}{2} \operatorname{shh} b_k + b_k \operatorname{chh} b_k & \operatorname{shh} b_k \\ -\operatorname{shh} b_k & \frac{a_k}{2} \operatorname{shh} b_k + b_k \operatorname{chh} b_k \end{bmatrix}, \tag{3.4}$$

$|a_k|=2$ esetén pedig

$$e^{hA_k} = e^{h\frac{a_k}{2}} \begin{bmatrix} 1-h\frac{a_k}{2} & h \\ -h & 1+h\frac{a_k}{2} \end{bmatrix} \tag{3.5}$$

$$\left(\text{itt } b_k = \sqrt{\left(\frac{a_k}{2}\right)^2 - 1}\right).$$

A főmátrix közelítő értéke tehát (3.4), ill. (3.5) típusú mátrixok szorzata. Ezek segítségével származtatható a karakterisztikus multiplikátorokat definiáló másodfokú polinom.

Mivel azonban

$$\det\left(e^{hA_{m-1}} \ldots e^{hA_0}\right) = \det e^{hA_{m-1}+\cdots+hA_0} \tag{3.6}$$

(jóllehet $A_k A_j \neq A_j A_k$, $j \neq k$), ezért csak

$$\det\left(e^{hA_{m-1}+\cdots+hA_0}\right) = e^{h\operatorname{Sp}(A_{m-1}+\cdots+A_0)} \tag{3.7}$$

számítandó. (3.7) így éppen annak a karakterisztikus multiplikátornak a közelítése, melyet meg akartunk határozni.

Legyen a továbbiakban

$$M = 2\max\left(1, \max_{t\in[0,\tau_0]} p|u_0^2(t)-1|\right),$$

ekkor nyilvánvaló, hogy $\|A(t)\|_1 \leqq \|A(t)\|_2 \leqq M$, továbbá $\|A_k\|_2 \leqq M$, $k=0, 1, 2, \ldots$ $\ldots, m-1$. $\varepsilon>0$-hoz m választható olyanra, hogy

$$\|A(t')-A(t'')\|_1 = p|u_0^2(t')-u_0^2(t'')| < \varepsilon \tag{3.8}$$

teljesüljön, ha $|t'-t''|<\delta(\varepsilon)\leqq\frac{\tau_0}{m}=h$. Ekkor viszont (1.10) felhasználásával a

$$\begin{aligned} |\det Y(\tau_0) - \det Y^h(\tau_0)| &\leqq \|Y(\tau_0)\|_2 \|Y(\tau_0) - Y^h(\tau_0)\|_2 < \\ &< 2e^{M\tau_0} 2\varepsilon\tau_0 e^{M(\tau_0+h)} = 4\varepsilon\tau_0 e^{M(2\tau_0+h)} \end{aligned} \tag{3.9}$$

becslés adódik az 1-től különböző karakterisztikus multiplikátor hibájára (itt $Y(\tau_0)$, ill. $Y^h(\tau_0)$ (3.2) főmátrixát, ill. főmátrixának (1.6) szerinti közelítését jelöli).

Vizsgáljunk meg egy háromdimenziós példát. Legyen adott az

$$\dddot{x}+a\ddot{x}+b\dot{x}+\tilde{f}(x)=0 \tag{3.10}$$

differenciálegyenlet, ahol $a>0$, $b>0$, $a^2<4b$, $\tilde{f}\in C^1_R$, 2π szerint periodikus, negatív, $\tilde{f}'(x)<ab$, továbbá, ha $\tilde{m}=\min(-f(x))$, $\tilde{m}+2d=\max(-f(x))$, akkor a

$$\frac{d}{\tilde{m}+d}<\frac{a\sqrt{4b-a^2}}{4b}$$

feltétel mellett (3.10)-nek van egy nem állandó D-periodikus megoldása (l. [4]): $u_0(t)$, $\tau_0>0$,

$$u_0(t)=\alpha_0(t)+a_0t,\quad \alpha_0(t+\tau_0)\equiv\alpha_0(t),\quad a_0\tau_0=2\pi.$$

(3.10)-nek $u_0(t)$-re vonatkozó variációs rendszere

$$\dot{y}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -\tilde{f}'_x(u_0(t)) & -b & -a\end{bmatrix}y,\quad y\in R^3. \tag{3.11}$$

Jelölje $f_k=\tilde{f}'_x(u_0(t_k))$,

$$A_k=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -f_k & -b & -a\end{bmatrix}.$$

(1.6) alkalmazásával közelíthető a főmátrix, ahol

$$e^{hA_k}\approx\begin{bmatrix}1 & h & \frac{h^2}{2}\\ -\frac{h^2}{2}f_k & 1-\frac{h^2}{2}b & h-\frac{h^2}{2}a\\ -hf_k+\frac{h^2}{2}af_k & -hb+\frac{h^2}{2}(ab-f_k) & 1-ha+\frac{h^2}{2}(a^2-b)\end{bmatrix}, \tag{3.12}$$

s e közelítés hibája

$$\frac{Mh^3e^{hM}}{6}$$

értékkel becsülhető, ahol

$$M=\max(1,\max(a,b,|\tilde{f}'_x(u_0(t))|)).$$

$\varepsilon>0$-hoz m választható olyanra, hogy

$$|\tilde{f}'_x(u_0(t'))-\tilde{f}'_x(u_0(t''))|<\varepsilon$$

legyen, ha $|t'-t''|<\delta(\varepsilon)\leqq h$, s ekkor

$$\delta(e^{hA_k})<\varepsilon he^{2Mh}+\frac{Mh^3}{6}e^{hM} \tag{3.13}$$

becslés adható e^{hA_k} hibakorlátjára. Ekkor

$$\|Y^h(\tau_0)-Y(\tau_0)\|_1 < \tau_0 e^{M\tau_0}\left(\varepsilon e^{Mh}+\frac{Mh^2}{6}\right), \tag{3.14}$$

ahol $Y(\tau_0)$, $Y^h(\tau_0)$ (3.11) főmátrixának pontos és közelítő értéke (3.12) szerint. Mivel a karakterisztikus egyenlet egyik gyöke az 1, a másik kettő kielégíti a

$$\varrho^2+(1+c_2)\varrho-c_0=0$$

másodfokú egyenletet. Itt $c_0=e^{a\tau_0}$ a *Liouville-formula* alapján ismert érték, tehát $\delta(c_0)$ 0-nak vehető. Így a karakterisztikus multiplikátorok hibáját $\delta(c_2)$ befolyásolja. Mivel

$$c_2=-\operatorname{Sp} Y^h(\tau_0),$$

ezért

$$\delta(c_2)=\delta(\operatorname{Sp} Y^h(\tau_0)) \leqq \|Y^h(\tau_0)-Y(\tau_0)\|_2 < 3\tau_0 e^{M\tau_0}\left(\varepsilon e^{Mh}+\frac{Mh^2}{6}\right). \tag{3.15}$$

Ebből (2.16) alkalmazásával a másik kettő karakterisztikus multiplikátor δ hibájára a

$$\delta < 3\tau_0 e^{M\tau_0}\left(\varepsilon e^{Mh}+\frac{Mh^2}{6}\right)+\sqrt{(1+|\operatorname{Sp} Y^h(\tau_0)|)\frac{3}{2}\tau_0 e^{M\tau_0}\left(\varepsilon e^{Mh}+\frac{Mh^2}{6}\right)}$$

becslés adható.

IRODALOM

[1] Демидович, Б. П., *Лекции по математической теории устойчивости* (Наука, Москва, 1967).

[2] FARKAS, M., "On isolated periodic solutions of differential systems", *Annali di Mat. pura appl.* **106** (1975) 233—243.

[3] Михайлов, Ф. А. и Орешников, В. Г., «Об одном классе достаточных условий асимптотической устойчивости периодических движений», (*Проблемы аналитической механики, теорий устойчивости и управления*, Наука, Москва, 1975).

[4] Назаров, Е. А., «Условия существования периодических траекторий в динамических системах с цилиндрическим фазовым пространством», *Дифф. Урав.*, **6** (1970) 337—380.

[5] PONTRJAGIN, L. SZ., *Közönséges differenciálegyenletek* (Akadémiai Kiadó, Budapest, 1972).

[6] RALSTON, A., *Bevezetés a numerikus analízisbe* (Műszaki Könyvkiadó, Budapest, 1969).

(Beérkezett: 1977. június 29.)

KOTSIS DOMOKOSNÉ
BME GÉPÉSZMÉRNÖKI KAR MATEMATIKA TANSZÉK
1111 BUDAPEST XI., STOCZEK U. H ÉP. IV EM.

THE APPROXIMATION OF THE CHARACTERISTIC MULTIPLIERS OF PERIODIC DIFFERENTIAL EQUATIONS

MRS. D. KOTSIS

In this paper the approximation of the characteristic multipliers of linear differential equations with periodic coefficients will be considered. The method is based upon a theorem of DEMIDOVICH. Algebraic conditions are given about the position of the characteristic multipliers on the complex plane together the estimation of the error of the approximative characteristic multipliers. Finally the effective determination of the characteristic multipliers is shown in two important cases.

Alkalmazott Matematikai Lapok **2** *(1976) 277—288*

STABILITÁSVIZSGÁLATOK VÉGES IDŐINTERVALLUMON DIFFERENCIÁLEGYENLŐTLENSÉGEK ALKALMAZÁSÁVAL

KLINCSIK MIHÁLY
Szeged

A dolgozatban megadjuk a véges időintervallumon való stabilitás elegendő feltételét LJAPUNOV második módszerének és a differenciálegyenlőtlenségek elméletének felhasználásával. A kapott feltétel általánosítását adja VAN DAN-CSZSI és SZ. J. SZTYEPANOV [3] eredményének. Ennek birtokában a nem-lineáris rendszerek véges időintervallumon való stabilitását az első közelítés stabilitásának vizsgálatára vezetjük vissza. A dolgozat eredményeit mechanikai mozgások stabilitásának vizsgálatával illusztráljuk.

1. Bevezetés

A múlt század végén A. M. LJAPUNOV [5] megadta a stabilitás egzakt matematikai definícióját, amely azt fejezi ki, hogy ha a kezdeti perturbáció elég kicsi, akkor a perturbált mozgás a perturbálatlantól tetszőlegesen kicsit tér el *a teljes mozgás során*, az időben egyenletesen. Sok gyakorlati problémánál egyrészt a mozgásokat csak véges ideig figyeljük. Másrészt a technikai feltételek megszabják, hogy a perturbált mozgás a perturbálatlantól mennyire térhet el. Ekkor a feladat: megállapítani a kezdeti perturbáció nagyságára vonatkozóan egy olyan becslést, amely biztosítja, hogy az adott intervallumon a fenti eltérés ne haladja meg az előírt mértéket. Ez a tartalma a *véges időintervallumon való stabilitás* fogalmának, amelyet N. G. CSETAJEV vezetett be 1960-ban [7]. A fogalommal, annak megjelenése óta ugyan több dolgozat és könyv is foglalkozott (ld. [4, 6]), de mégsem került olyan széles körben alkalmazásra, mint ahogyan azt várni lehetett volna. Ez valószínűleg annak a következménye, hogy elég kevés olyan feltétel született, amely a gyakorlatban is könnyen alkalmazható. A leghatékonyabbnak és az alkalmazások számára is legkönnyebben hozzáférhetőnek a véges időintervallumon való stabilitás tanulmányozására is LJAPUNOV második módszere bizonyult. VAN DAN-CSZSI és SZ. J. SZTYEPANOV 1975-ben ezzel a módszerrel elegendő feltételt adtak a véges időintervallumon való stabilitásra. Dolgozatunk 3. pontjában a differenciálegyenlőtlenségek alaptételének felhasználásával ezt a feltételt sikerült lényegesen továbbfejlesztenünk. A 3. pont eredményeinek felhasználásával a 4. pontban feltételt vezetünk le a lineáris egyenletrendszer véges időintervallumon való stabilitására és instabilitására. Ismeretes, hogy a *Ljapunov-féle stabilitási elmélet*ben is központi helyet foglalnak el azok a tételek, amelyek azt vizsgálják, hogy nem-lineáris rendszerekre mikor lehet a stabilitást a lineáris közelítés vizsgálatával eldönteni. Dolgozatunk 5. pontjában ezzel a kérdéssel foglalkozunk a véges időintervallumon való stabilitás esetében. Eredményeinket többek között a változó fonalhosszúságú matematikai inga és a centrifugális regulátor példájával illusztráljuk.

A szerző itt köszöni meg HATVANI LÁSZLÓnak a probléma felvetését és a dolgozat elkészítésében nyújtott segítségét.

2. Alapvető fogalmak és definíciók

Legyen adott a perturbált mozgások

$$\dot{x} = X(t, x), \quad X(t, 0) \equiv 0 \tag{2.1}$$

differenciálegyenlet-rendszere, ahol $x=(x_1, \ldots, x_n)^T$ az R^n n-dimenziós valós euklideszi tér egy oszlopvektorát jelöli. Az $\|x\| = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2}$ az x vektor normáját jelenti. Feltesszük, hogy (2.1) jobb oldala értelmezett és folytonos a

$$\Gamma = \{(t, x): t \geqq 0, \|x\| < H\} \quad (0 < H = \text{konst.})$$

halmazon, továbbá, hogy bármely $(t_0, x_0) \in \Gamma$ értékre (2.1)-nek pontosan egy $x(t; t_0, x_0)$-val jelölt megoldása létezik, mely az $x(t_0; t_0, x_0) = x_0$ kezdeti feltételt elégíti ki. A megoldásokról feltesszük, hogy egy előre adott $[t_0, T]$ véges időintervallumon léteznek.

1. DEFINÍCIÓ: Legyenek λ, A adott valós számok, amelyekre $0 < \lambda < A$ teljesül. A (2.1) rendszer $x=0$ megoldását (λ, A, t_0, T)-*stabilisnak* nevezzük, ha minden $\|x_0\| \leqq \lambda$ esetén az $\|x(t; t_0, x_0)\| < A$ egyenlőtlenség teljesül a $t \in [t_0, T]$ intervallumon.

2. DEFINÍCIÓ: Azt mondjuk, hogy a (2.1) rendszer 0-megoldása *egyenlő mértékben* (λ, A, t_0, T)-*stabilis*, ha minden $t_1 \in [t_0, T]$ időpontra a (2.1) 0-megoldása (λ, A, t_1, T)-stabilis.

Nyilvánvaló, hogy ha (2.1) jobb oldala nem függ t-től, vagyis a rendszer autonóm, akkor a két fogalom ekvivalens.

Ezeket a fogalmakat N. G. CSETAJEV [7] vezette be 1960-ban. Mint ismeretes, a *stabilitás* klasszikus, A. M. LJAPUNOVtól [5] származó definíciója azt kívánja meg, hogy bármely $\varepsilon > 0$ és t_0-hoz létezzen $\delta(t_0, \varepsilon) > 0$ úgy, hogy ha $\|x_0\| \leqq \delta(t_0, \varepsilon)$ akkor az $\|x(t; t_0, x_0)\| < \varepsilon$, az egész $t \in [t_0, +\infty)$ időintervallumon. A lényeges eltérés a két fogalom között tehát abban van, hogy a véges időintervallumon való stabilitás esetében az A és λ (melyek ε-nak és δ-nak felelnek meg) előre adottak és az időintervallum, melyen az eltérést vizsgáljuk, véges.

3. A véges időintervallumon való stabilitás és Ljapunov második módszere

Megadjuk a véges időintervallumon való stabilitás ill. egyenlő mértékben való stabilitás elegendő kritériumát LJAPUNOV második módszere és a differenciálegyenlőtlenségek WAŻENSKI-féle [2] alaplemmája segítségével.

Legyen $z(t)$ folytonos skalár függvény. Ha létezik a

$$D_- z(t) = \liminf_{h \to +0} \left(z(t) - z(t-h)\right)/h,$$

akkor ezt a $z(t)$ függvény *Dini-féle alsó baloldali deriváltjának* nevezzük.

Ważewski-lemma: Legyen az $f(t, y)$ skalárfüggvény értelmezett és folytonos az $[a, b] \times R^1$ halmazon és $y(t)$ az $\dot{y} = f(t, y)$, $y(a) = y_0$ kezdetiérték-probléma maximális megoldása az $[a, b]$ intervallumon. Ha a $z(t)$ folytonos skalárfüggvény kielégíti a $z(a) \leqq y_0$ egyenlőtlenséget és $D_- z(t) \leqq f(t, z(t))$ a $t \in [a, b]$ intervallumon, akkor $z(t) \leqq y(t)$ minden $t \in [a, b]$ értékre.

A továbbiakban $V(t, x)$-szel olyan, Γ-n értelmezett skalár függvényt jelölünk, melynek léteznek a folytonos parciális deriváltjai a Γ-n (kivéve esetleg a $(t, 0)$ félegyenest). Az ilyen függvényt *Ljapunov-függvény*nek nevezzük. A $V(t, x)$ függvény (2.1) rendszer szerinti deriváltja a

$$\dot{V}(t, x) = \frac{\partial V(t, x)}{\partial t} + \sum_{i=1}^{n} X_i(t, x) \cdot \frac{\partial V(t, x)}{\partial x_i} \tag{3.1}$$

skalár függvény.

3.1. TÉTEL. Tegyük fel, hogy létezik olyan $V(t, x)$ *Ljapunov-függvény* és $f(t, y)$, a $[t_0, T] \times R^1$ halmazon értelmezett, folytonos skaláris függvény, melyek teljesítik az alábbi feltételeket:

(i) $\lambda' = \max_{\|X\| \leqq \lambda} V(t_0, x) < A' = \min_{\substack{\|X\| = A \\ t \in [t_0, T]}} V(t, x)$;

(ii) $\dot{V}(t, x) \leqq f(t, V(t, x))$ a $D = \{(t, x): t \in [t_0, T], \|x\| \leqq A\}$ halmazon;

(iii) az $\dot{y} = f(t, y)/f(t, 0) \equiv 0/$ skaláris differenciálegyenlet $y = 0$ megoldása (λ', A', t_0, T)-stabilis.

Akkor (2.1) 0-megoldása (λ, A, t_0, T)-stabilis.

Bizonyítás. Indirekt úton bizonyítunk, tegyük fel, hogy a (2.1) rendszer 0-megoldása nem (λ, A, t_0, T)-stabilis, azaz van olyan x_0 és t^*, hogy $\|x_0\| \leqq \lambda$, $t^* \in [t_0, T]$, mégis $\|x(t^*; t_0, x_0)\| = A$. Az $x(t; t_0, x_0)$ folytonossága miatt feltehető, hogy t^* az első ilyen hely, vagyis $\|x(t; t_0, x_0)\| < A$ minden $t \in [t_0, t^*)$ időpontra. Így az $x(t; t_0, x_0)$ megoldás a $[t_0, t^*]$ intervallumon a D halmazon belül marad, ahol (ii) érvényes, ezért $\dot{V}(t, x(t)) \leqq f(t, V(t, x(t)))$ a $t \in [t_0, t^*]$-ra. A *Ważewski-lemma* alapján az

$$\dot{y} = f(t, y) \tag{3.2}$$

differenciálegyenlet $y(t_0) = y_0 = V(t_0, x_0)$ kezdeti feltételt kielégítő $y(t; t_0, y_0)$ maximális megoldására $V(t, x(t; t_0, x_0)) \leqq y(t; t_0, y_0)$ a $t \in [t_0, t^*]$ intervallumon. Mivel $|y_0| = |V(t_0, x_0)| \leqq \lambda'$ és a (3.2) 0-megoldása (λ', A', t_0, T)-stabilis, $y(t^*; t_0, y_0) < A'$. Tehát

$$A' = \min_{\substack{\|X\| = A \\ t \in [t_0, T]}} V(t, x) \leqq V(t^*, x(t^*)) \leqq y(t^*; t_0, y_0) < A'.$$

A kapott ellentmondás az állítást igazolja.

3.2. TÉTEL. Tegyük fel, hogy létezik olyan $V(t, x)$ *Ljapunov-függvény* és $f(t, y)$, a $[t_0, T] \times R^1$-en értelmezett, folytonos skaláris függvény, melyek kielégítik az alábbi feltételeket:

(i) $\lambda = \max_{\substack{\|X\| = \lambda \\ t \in [t_0, T]}} V(t, x) < A' = \min_{\substack{\|X\| = A \\ t \in [t_0, T]}} V(t, x)$;

(ii) $\dot{V}(t, x) \leqq f(t, V(t, x))$

a $H = \{(t, x): t \in [t_0, T], \lambda \leqq \|x\| \leqq A\}$ halmazon;

(iii) az $\dot{y}=f(t, y)/f(t, 0)\equiv 0/$ skaláris differenciálegyenlet 0-megoldása egyenlő mértékben $(\bar{\lambda}, A', t_0, T)$-stabilis.

Akkor (2.1) 0-megoldása egyenlő mértékben (λ, A, t_0, T)-stabilis.

Bizonyítás. Indirekt úton bizonyítunk, tegyük fel, hogy a feltételek teljesülnek, mégis van olyan x_0 és t_0', hogy az $\|x_0\|\leqq\lambda$, $t_0'\in[t_0, T]$ és az $x(t; t_0', x_0)$ megoldás az $x=0$ megoldástól normában legalább A-val tér el. Az $x(t; t_0', x_0)$ megoldás folytonossága miatt léteznek olyan t_1, t_2 időpontok, hogy $t_0'\leqq t_1<t_2\leqq T$ és $\|x(t_1; t_0', x_0)\|=\lambda$, $\|x(t_2; t_0', x_0)\|=A$, továbbá minden $t\in(t_1, t_2)$ időpontra $\lambda<\|x(t; t_0', x_0)\|<A$ teljesül. Tehát ez a megoldás a $[t_1, t_2]$ intervallumon a H halmazon belül marad, ahol (ii) alapján $\dot{V}(t, x(t))\leqq f(t, V(t, x(t)))$ $(t\in[t_1, t_2])$.

Az $\dot{y}=f(t, y)$ differenciálegyenlet $y(t_1)=y_1=V(t_1, x(t_1))$ kezdeti feltételnek eleget tevő $y(t; t_1, y_1)$ maximális megoldására $|y(t; t_1, y_1)|<A'$ $(t\in[t_1, t_2])$ a (iii) feltétel szerint, hiszen $|y_1|\leqq\bar{\lambda}$.

A *Ważewski-lemma* felhasználásával kapjuk, hogy

$$V(t, x(t; t_0', x_0))\leqq y(t; t_1, y_1)\quad (t\in[t_1, t_2]).$$

A $t=t_2$ helyen az egyenlőtlenség

$$A'=\min_{\substack{\|X\|=A\\ t\in[t_0, T]}} V(t, x)\leqq V(t_2, x(t_2))\leqq y(t_2; y_1, t_1)<A'.$$

A kapott ellentmondás az állítást igazolja.

VAN DAN-CSZSI és SZ. J. SZTYEPANOV [3] 1975-ben megadták a stabilitás és egyenlő mértékben való stabilitás jellemzését LJAPUNOV második módszerével. Eredményeiket tételeink az $f(t, y)\equiv 0$ esetként tartalmazzák. A következő példa mutatja, hogy a 3.1. és 3.2. tétel VAN DAN-CSZSI és SZTYEPANOV eredményeinek lényeges általánosításai.

3.1. PÉLDA. Tekintsük a következő mozgásegyenletet:

$$\ddot{y}+a(t)\dot{y}+b(t)g(y)=0, \tag{3.3}$$

ahol $a(t)$ folytonos, $b(t)>0$ folytonosan differenciálható a $[t_0, T]$ intervallumon. A $g(y)$ függvény folytonos $y\in(-\infty, +\infty)$ esetén, $g(0)=0$ és $\int_0^{x_1} g(s)\,ds>0$, ha $x_1\neq 0$. Legyen $x_1=y$, $x_2=\dot{y}$, ekkor a (3.3) differenciálegyenlet ekvivalens az

$$\begin{aligned}\dot{x}_1&= x_2\\ \dot{x}_2&=-b(t)\cdot g(x_1)-a(t)\cdot x_2\end{aligned}$$

rendszerrel. A $V(t, x)$ *Ljapunov-függvény* legyen

$$V(t, x_1, x_2)=\frac{x_2^2}{b(t)}+2\cdot\int_0^{x_1} g(s)\,ds \tag{3.4}$$

alakú. Mivel $\dot{V}=-x_2^2(\dot{b}(t)/b(t)+2a(t))/b(t)$ a VAN DAN-CSZSI és SZTYEPANOV

$\dot{V} \leqq 0$ feltétele nem teljesül ilyen általános esetben. Ha $c \in R^1$ valós szám, akkor jelölje

$$[c]_+ = \max\{0, c\}, \quad [c]_- = \min\{0, -c\}$$

c pozitív, illetve *negatív részét.* Ezek alapján a *Ljapunov-függvény*-rendszer szerinti deriváltja a $\dot{V} \leqq [\dot{b}(t)/b(t) + 2a(t)]_- V$ egyenlőtlenségnek tesz eleget. A 3.1. tétel alapján (3.3) egyenlet $y = \dot{y} = 0$ megoldásának (λ, A, t_0, T)-stabilitásához az $\dot{y} = [\dot{b}(t)/b(t) + 2a(t)]_- \cdot y$ lineáris differenciálegyenlet 0-megoldásának stabilitását kell megvizsgálni, amit a megoldás megkeresésével dönthetünk el.

3.2. PÉLDA. Tekintsünk egy m tömegű matematikai ingát, amelynek l hossza az időben változik az előre adott $l = l(t)$ törvény szerint $(l(t) > 0)$. A rendszer kinetikai energiája $T = 1/2m\,[l^2(t)\dot{\varphi}^2 + (\dot{l}(t))^2]$, potenciális energiája $V = mgl(t)\,(\cos\varphi - 1)$, ahol φ az ingának a függőlegessel bezárt szögét jelöli. Az $L = T - V$ *Lagrange-függvény* segítségével és a

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\varphi}}\right) - \frac{\partial L}{\partial \varphi} = 0.$$

*Lagrange-féle másodfajú mozgásegyenlet*ből a

$$\ddot{\varphi} + 2\frac{\dot{l}(t)}{l(t)}\dot{\varphi} + \frac{g}{l(t)}\sin\varphi = 0 \tag{3.5}$$

mozgásegyenletet kapjuk.

Vizsgáljuk meg az $l(t) = 2 - \cos t$ periodikus törvény szerint változó fonalhosszúságú inga $(\lambda, A, 0, 2\pi)$-stabilitását a (3.4) alakú $V(t, \varphi, \dot{\varphi}) = l(t)\dot{\varphi}^2/g + 2(1 - \cos\varphi)$ *Ljapunov-függvénny*el. Ekkor

$$\dot{V} = -\frac{\dot{l}(t)}{g}\dot{\varphi}^2\left(3\frac{\dot{l}(t)}{l(t)}\right) \Rightarrow \dot{V} \leqq 3\left[\frac{\dot{l}(t)}{l(t)}\right]_- V.$$

A 3.1. tétel alapján tehát az $\dot{y} = 3[\sin t/(2 - \cos t)]_- y$ differenciálegyenlet stabilitásának vizsgálatára vezettük vissza a feladatot, melynek általános megoldása

$$y(t) = \begin{cases} y(0), & \text{ha} \quad t \in [0, \pi], \\ y(0) \cdot \left(\dfrac{2 - \cos\pi}{2 - \cos t}\right)^3, & \text{ha} \quad t \in [\pi, 2\pi]. \end{cases}$$

A $[0, \pi]$ intervallumon a megoldás konstans, a $[\pi, 2\pi]$-n pedig

$$|y(t)| = |y(0)|\frac{27}{|\cos t - 2|^3} \leqq 27\lambda'.$$

Adott A esetén a λ-értékét úgy kell meghatározni, hogy a $\lambda' < A'/27$ egyenlőtlenség teljesüljön, ahol a λ', A' értékek a

$$\lambda' = \max_{\varphi^2 + \dot{\varphi}^2 = \lambda^2} V(\pi, \varphi, \dot{\varphi}) = \max_{0 \leqq v \leqq 2\pi}\left(0{,}3\lambda^2 \sin^2 v + 4\sin^2\left(\frac{\lambda}{2}\cos v\right)\right)$$

és

$$A' = \min_{\substack{\varphi^2 + \dot{\varphi}^2 = A^2 \\ t \in [\pi, 2\pi]}} V(t, \varphi, \dot{\varphi}) = \min_{0 \leqq v \leqq 2\pi}\left(0{,}1A^2 \sin^2 v + 4\sin^2\left(\frac{A}{2}\cos v\right)\right)$$

képletekből határozhatók meg. Az $A=10^{-3}$ értékhez a fenti formulák felhasználásával számítógép segítségével $\lambda=0{,}6\cdot 10^{-4}$ értéket kapunk.

3.3. PÉLDA. Tekintsük az

$$\ddot{x}+a(t)x=0 \quad (a(t)>0,\ x\in\mathbf{R}^1) \tag{3.6}$$

*Jacobi-egyenlet*et a speciális $a(t)=2-\cos t$ esetben a $[0, 2\pi]$ intervallumon. Legyen a *Ljapunov-függvény* $V(t, x, \dot{x})=\dot{x}^2/2+a(t)x^2/2$ alakú és ekkor a (3.6) szerinti deriváltja:

$$\dot{V}=\frac{\dot{a}(t)}{2}x^2=\frac{\dot{a}(t)}{a(t)}\left(V-\frac{\dot{x}^2}{2}\right)\leqq\left[\frac{\dot{a}(t)}{a(t)}\right]_+V.$$

A (3.1) tétel értelmében elegendő a

$$\dot{y}=\left[\frac{\dot{a}(t)}{a(t)}\right]_+y \tag{3.7}$$

differenciálegyenlet $(\lambda', A', 0, 2\pi)$-stabilitását vizsgálni, ahol

$$\lambda'=\max_{x^2+\dot{x}^2=\lambda^2}V(0,x,\dot{x})=\lambda^2/2 \quad \text{és} \quad A'=\min_{\substack{x^2+\dot{x}^2=A^2\\ t\in[0,\pi]}}V(t,x,\dot{x})=A^2/2.$$

A (3.7) általános megoldása az $a(t)=2-\cos t$ esetben

$$y(t)=\begin{cases} y(0)\dfrac{2-\cos t}{2-\cos 0}, & \text{ha} \quad t\in[0,\pi];\\ y(\pi), & \text{ha} \quad t\in[\pi,2\pi]. \end{cases}$$

Most a $[\pi, 2\pi]$ intervallumon konstans a megoldás, a $[0, \pi]$-n pedig $|y(t)|\leqq$ $\leqq\lambda'(2-\cos t)\leqq 3\lambda'$. És így a $\lambda'<A'/3$ teljesülése esetén, azaz $\lambda<A/\sqrt{3}$ mellett, a (3.6) 0-megoldása $(\lambda, A, 0, 2\pi)$-stabilis.

4. Lineáris differenciálegyenletrendszerek stabilitása

Ebben a pontban az

$$\dot{x}=A(t)x \tag{4.1}$$

lineáris differenciálegyenletrendszer (λ, A, t_0, T)-stabilitásának egy szükséges és egy elegendő feltételét adjuk meg Ljapunov második módszerével. Az $A(t)$ a $[t_0, T]$ intervallumon értelmezett és folytonos $n\times n$-es mátrixot jelöl, továbbá jelentse $\Omega(t)(\omega(t))$ az $(A(t)+A^T(t))/2$ mátrix legnagyobb (legkisebb) sajátértékét a $t\in[t_0, T]$ időpontban és $A^T(t)$ az $A(t)$ mátrix transzponáltját.

4.1. LEMMA. Ha fennáll az

$$\ln\frac{A}{\lambda}>\int_{t_0}^{t}\Omega(s)\,ds \tag{4.2}$$

egyenlőtlenség a $t\in[t_0, T]$ intervallumon, akkor (4.1) 0-megoldása (λ, A, t_0, T)-stabilis.

(ii) Ha létezik olyan $t_1 \in [t_0, T]$ időpont, melyre fennáll az

$$\ln \frac{A}{\lambda} \leqq \int_{t_0}^{t_1} \omega(s)\, ds \tag{4.3}$$

egyenlőtlenség, akkor (4.1) 0-megoldása (λ, A, t_0, T)-instabilis, azaz nem (λ, A, t_0, T)-stabilis.

Bizonyítás. Legyen a *Ljapunov-függvény* $V(x) = \|x\|^2$. A (4.1) rendszer szerinti deriváltja

$$\dot{V}(x) = (A(t)x, x) + (x, A(t)x) = 2\left(x, \frac{A(t)+A^T(t)}{2}x\right),$$

ahol $(x, y) = \sum_{i=1}^{n} x_i y_i$ az $x, y \in \mathbf{R}^n$ vektorok skalár szorzatát jelöli. A kvadratikus alakok szélsőérték tulajdonsága alapján

$$2\omega(t)\|x\|^2 \leqq \dot{V} \leqq 2\Omega(t)\|x\|^2.$$

(i) A $\dot{V} \leqq 2\Omega(t)V$ egyenlőtlenség és a 3.1. tétel alapján a (4.1) 0-megoldásának (λ, A, t_0, T)-stabilitásához elegendő a

$$\dot{y} = 2\Omega(t)y \tag{4.4}$$

differenciálegyenlet (λ^2, A^2, t_0, T)-stabilitását teljesíteni. (4.4) általános megoldása $y(t) = y(t_0)\exp\left(2\int_{t_0}^{t} \Omega(s)ds\right)$ alakú. Ha $|y(t_0)| \leqq \lambda^2$ és teljesül a (4.2) feltétel, akkor

$$|y(t)| < \lambda^2 \exp\left(2\ln\frac{A}{\lambda}\right) = A^2.$$

Tehát valóban (4.1) 0-megoldása (λ, A, t_0, T)-stabilis.

(ii) Feltehetjük, hogy $x(t)$ nem a triviális megoldás, és ekkor a triviális megoldás unicitása miatt $x(t) \neq 0$ $(t \in [t_0, T])$, ezért a

$$2\omega(t) \leqq \frac{\dot{V}(x(t))}{V(x(t))}$$

alapján $2 \cdot \int_{t_0}^{t_1} \omega(s)ds \leqq \ln \frac{V(x(t_1))}{V(x(t_0))}$ teljesül. Ha a t_1 kielégíti a (4.3) feltételt és $\|x_0\| = \lambda$, akkor $V(x)$ definíciója alapján

$$\|x(t_1)\|^2 \geqq \|x(t_0)\|^2 \exp\left(2\int_{t_0}^{t_1} \omega(s)\, ds\right) \geqq \lambda^2 \exp\left(2\ln\frac{A}{\lambda}\right) = A^2.$$

Ahonnan $\|x(t_1)\| \geqq A$ következik, és így valóban (4.1) 0-megoldása (λ, A, t_0, T)-instabilis.

4.1. KÖVETKEZMÉNY. a) Ha a (4.1) 0-megoldása (λ, A, t_0, T)-stabilis, akkor

$$\ln \frac{A}{\lambda} > \int_{t_0}^{t} \omega(s)\, ds \quad (t \in [t_0, T]). \tag{4.5}$$

b) Ha az $(A(t)+A^T(t))/2$ mátrix legnagyobb sajátértéke is negatív a $[t_0, T]$ intervallumon, akkor a (4.1) rendszer stabilis tetszőleges $0<\lambda<A$ érték mellett.

c) Legyen az $A(t)$ mátrix konstans, ekkor

ha az $\Omega<\frac{1}{T-t_0}\ln\frac{A}{\lambda}$ egyenlőtlenség teljesül, akkor (4.1) 0-megoldása (λ, A, t_0, T)-stabilis;

ha (4.1) 0-megoldása (λ, A, t_0, T)-stabilis, akkor teljesül az $\omega<\frac{1}{T-t_0}\ln\frac{A}{\lambda}$ egyenlőtlenség.

4.1. PÉLDA. A 2.3. példában tekintett

$$\ddot{x}+(2-\cos t)x = 0$$

Jacobi-egyenletre

$$\frac{A(t)+A^T(t)}{2} = \begin{bmatrix} 0 & \frac{\cos t-1}{2} \\ \frac{\cos t-1}{2} & 0 \end{bmatrix}.$$

Így $\Omega(t)=\frac{1-\cos t}{2}$ és $\omega(t)=\frac{\cos t-1}{2}$ lesz. A 4.1. lemma alapján, ha a

$$\sup_{t\in[0,2\pi]}\frac{1}{2}(t-\sin t) = \pi < \ln\frac{A}{\lambda}$$

teljesül, akkor a *Jacobi-egyenlet* 0-megoldása (λ, A, t_0, T)-stabilis, amit összehasonlítva a (3.3) példában kapott eredménnyel láthatjuk, hogy a $V(t, x, \dot{x})=\dot{x}^2/2+$ $+(2-\cos t)x^2/2$ *Ljapunov-függvénnyel* kapott eredmény jobb, mert adott A esetén nagyobb λ értékeket is megenged.

4.1. *Megjegyzés.* A 4.1. lemma eredményei, a *Ważewski-egyenlőtlenség* [2] alapján is levezethetők.

5. Nem-lineáris differenciálegyenletrendszerek stabilitása első közelítés alapján

Legyen a differenciálegyenletrendszer az

$$\dot{x} = A(t)x+R(t, x) \tag{5.1}$$

alakban adva, ahol az

$$\|R(t, x)\| \leqq K\cdot\|x\| \quad (0 < K = \text{konst.}, (t, x)\in D) \tag{5.2}$$

feltétel teljesül. Az (5.1) rendszer közelítésén az

$$\dot{x} = A(t)x \tag{5.3}$$

rendszert értjük. Mint ahogyan a 4. pontban is $\Omega(t)$ (ill. $\omega(t)$) az $(A(t)+A^T(t))/2$ mátrix legnagyobb (ill. legkisebb) sajátértékét jelöli.

Azt vizsgáljuk, hogy a (5.3) közelítés mely stabilitási tulajdonságaiból következtethetünk az eredeti (5.1) rendszer stabilitására.

5.1. TÉTEL. i) Ha fennáll a

$$K(t-t_0)+\int_{t_0}^{t}\Omega(s)\,ds < \ln\frac{A}{\lambda} \tag{5.4}$$

egyenlőtlenség a $t\in[t_0, T]$ intervallumon, akkor (5.1) 0-megoldása (λ, A, t_0, T)-stabilis.

ii) Ha létezik olyan $t_1\in[t_0, T]$ időpont, amelyre a

$$-K(t_1-t_0)+\int_{t_0}^{t_1}\omega(s)\,ds \geqq \ln\frac{A}{\lambda} \tag{5.5}$$

egyenlőtlenség teljesül, akkor (5.1) 0-megoldása (λ, A, t_0, T)-instabilis.

Bizonyítás. Tekintsük a $V(x)=(x, x)=\|x\|^2$ *Ljapunov-függvényt*. Az (5.1) rendszer szerinti deriváltja

$$\dot{V}=(A(t)x+R(t, x), x)+(x, A(t)x+R(t, x))=2\cdot\left\{\left(x, \frac{A(t)+A^T(t)}{2}x+R(t, x)\right)\right\}.$$

A kvadratikus alakok szélsőérték-tulajdonsága alapján

$$\omega(t)\|x\|^2 \leqq \left(x, \frac{A(t)+A^T(t)}{2}x\right) \leqq \Omega(t)\|x\|^2$$

és a *Bunyakovszkij—Schwartz-féle egyenlőtlenség* szerint a

$$-\|R(t, x)\|\cdot\|x\| \leqq (x, R(t, x)) \leqq \|R(t, x)\|\cdot\|x\|$$

teljesül. Az (5.2) feltételt felhasználva tehát

$$2(\omega(t)-K)\|x\|^2 \leqq \dot{V} \leqq 2(\Omega(t)+K)\|x\|^2 \quad ((t, x)\in D).$$

i) Mivel a $\dot{V}\leqq 2(\Omega(t)+K)\cdot V$ egyenlőtlenség teljesül a D halmazon, ezért a 3.1. tétel értelmében az $\dot{y}=2(\Omega(t)+K)y$ differenciálegyenlet 0-megoldásának (λ^2, A^2, t_0, T)-stabilitását elegendő megvizsgálni. Ennek általános megoldása

$$y(t)=y(t_0)\exp\left(2\left(K(t-t_0)+\int_{t_0}^{t}\Omega(s)\,ds\right)\right).$$

Ha $|y(t_0)|\leqq\lambda^2$ és az (5.4) feltétel teljesül, akkor

$$|y(t)| < \lambda^2\exp\left(2\ln\frac{A}{\lambda}\right)=A^2 \quad (t\in[t_0, T]).$$

Így az (5.4) feltétel teljesülése esetén (5.1) 0-megoldása valóban (λ, A, t_0, T)-stabilis.

ii) Feltehetjük, hogy $x(t)$ nem a triviális megoldás és ekkor a triviális megoldás unicitása miatt $x(t)\neq 0$ $(t\in[t_0, T])$ ezért, a

$$2(\omega(t)-K) \leqq \frac{\dot{V}(x(t))}{V(x(t))}$$

egyenlőtlenségből kapjuk a

$$2\left(-K(t_1-t_0)+\int_{t_0}^{t_1}\omega(s)\,ds\right)\leqq \ln\frac{\|x(t_1)\|^2}{\|x(t_0)\|^2}$$

egyenlőtlenséget. Legyen $\|x(t_0)\|=\lambda$ és teljesüljön t_1-re az (5.5) feltétel, ekkor

$$\|x(t_1)\|^2\geqq\|x(t_0)\|^2\exp\left(2\left(-K(t_1-t_0)+\int_{t_0}^{t_1}\omega(s)\,ds\right)\right)\geqq A^2.$$

Vagyis (5.5) teljesülése esetén $\|x(t_1)\|\geqq A$, azaz (5.1) 0-megoldása (λ, A, t_0, T)-instabilis.

5.1. KÖVETKEZMÉNY. Az 5.1. tétel ii) állításából a stabilitás szükséges feltétele következik:

Ha az (5.1) 0-megoldása (λ, A, t_0, T)-stabilis, akkor

$$-K(t-t_0)+\int_{t_0}^{t}\omega(s)\,ds<\ln\frac{A}{\lambda}\quad(t\in[t_0, T]).$$

5.1. *Megjegyzés.* Ha az (5.3) lineáris rendszer az (5.1) rendszer első közelítése, vagyis az (5.2) helyett

$$\|R(t, x)\|\leqq c\|x\|^{1+\alpha}\quad(0<\alpha=\text{konst.},\ c=\text{konst.})$$

teljesül a D halmazon, akkor az (5.2) becslés teljesül tetszőleges kicsiny K-val, ha az A értéke elég kicsi. Tehát, ha a 4.1 lemma (4.2) feltétele teljesül, akkor az 5.1. tétel (5.4) feltétele is teljesül elég kicsi A-val. Ezt úgy is kifejezhetjük, hogy ha az (5.1) rendszer első közelítésének 0-megoldása úgy (λ, A, t_0, T)-stabilis, hogy (4.2) is teljesül, akkor az eredeti rendszer 0-megoldása is (λ, A, t_0, T)-stabilis.

5.1. PÉLDA. Az (5.4) formula lehetőséget ad arra, hogy első közelítés alapján egy adott rendszer (λ, A, t_0, T)-stabilitásának kérdését számítógép segítségével döntsük el. A gyakorlatban általában előre adott az A értéke, valamint az időintervallum és ehhez a lehető legnagyobb λ értéket kell meghatározni, melyre a stabilitás még teljesül.

A programot FORTRAN—IV nyelven megírtuk, mely az $(A(t)+A^T(t))/2$ szimmetrikus mátrix legnagyobb sajátértékét *Jacobi-módszer*rel, az integrált *Simpson-formula* alapján határozza meg. A 3.2. példában vizsgált változó fonalhosszúságú matematikai inga esetében a $[0, 2\pi]$ intervallumon $A=10^{-3}$ értékhez a $\lambda\leqq$ $\leqq 0{,}2382\cdot 10^{-9}$ értéket kaptunk.

5.2. PÉLDA. Az automatikus szabályozások elméletében igen fontos centrifugális szabályozó stabilitását is első közelítés alapján vizsgáltuk. Ez egy olyan automatikus szabályozó eszköz, mely a gőzgép kerekének szögsebességét adott értékre stabilizálja, a gőz nyomásától függetlenül (ld. [1]).

Ha az ω szögsebességgel forgó és J tehetetlenségi nyomatékú kerék a szabályozóhoz n áttételi számú fogaskerékrendszeren keresztül csatlakozik, továbbá

φ a szabályozó forgását jellemző nyílásszög, akkor a mozgást leíró differenciálegyenletrendszer

$$\dot{\varphi} = \psi$$
$$\dot{\psi} = n^2\omega^2 \sin\varphi \cdot \cos\varphi - g \cdot \sin\varphi - \frac{b}{m}\psi$$
$$\dot{\omega} = \frac{k}{J}\cos\varphi - \frac{F}{J},$$

ahol F a terheléstől függő mennyiség, $k>0$ arányossági tényező, $b>0$ súrlódási együttható, m a szabályozón levő tömeg. A $\psi_0=0$, $\cos\varphi_0=F/k$ és $n^2\omega_0^2=g/\cos\varphi_0$ egyensúlyi helyzet stabilitásának feltételeit vizsgáljuk. A $\varphi=\varphi_0+x_1$, $\psi=x_2$, $\omega=\omega_0+x_3$ új változók bevezetésével az

$$(5.6)\quad \dot{x} = Bx + R(x), \quad B = \begin{bmatrix} 0 & 1 & 0 \\ -\dfrac{g\sin^2\varphi_0}{\cos\varphi_0} & -\dfrac{b}{m} & \dfrac{2g\sin\varphi_0}{\omega_0} \\ -\dfrac{k}{J}\sin\varphi_0 & 0 & 0 \end{bmatrix}, \quad \|R\| \leqq K\cdot\|x\|^2,$$

rendszer $x=0$ megoldása stabilitásának vizsgálatára vezettük vissza a feladatot, ahol

$$(5.7)\quad K = 1/2\left\{\frac{k^2}{J^2} + \max^2\left(g + 2n^2(\omega_0+A)(1+\omega_0+A),\ n^2 + 2n^2(\omega_0+A)\right)\right\}^{1/2}.$$

Vizsgáljuk az (5.6) rendszer 0-megoldásának (λ, A, 0, 10)-stabilitását az (5.4) formula felhasználásával. Adott A-hoz akkor tudunk minél nagyobb λ-t meghatározni, ha a K értéke minél kisebb. Így az (5.7) összefüggésből világos, hogy a J tehetetlenségi nyomaték csökkentése a stabilitásra károsan hat, mely VISNYEGRADSZKIJ [1] vizsgálatainak eredményével egyezik. Olyan tehetetlenségi nyomaték mellett, melyre a rendszer *Ljapunov-szerint „erősen" instabilis*, az (5.4) formulával $A=10^{-3}$ értékhez a λ-ra gépi nullát kapunk, mely azt jelenti, hogy a rendszer gyakorlatilag instabilis. *Ljapunov-stabilis* adatok mellett az $A=10^{-3}$ értékhez $\lambda=0{,}1109\cdot10^{-10}$-t kaptunk.

IRODALOM

[1] PONTRJAGIN, L. Sz., *Közönséges differenciálegyenletek* (Akadémiai Kiadó, Budapest, 1972).
[2] WALTER, W., *Differential and Integral Inequalities* (Springer Verlag Berlin, Heidelberg, New York, 1970).
[3] Ван Дань-чжи и Степанов, С. Я., «Устойчивость на конечном интервале времени и ее численное исследование», в сб.: *Задачи исследования устойчивости и стабилизации движения*, . Вычислительный центр АН СССР 1 (1975) 3—58.
[4] Карачаров, К. А. и Пилютик, А. Г., *Введение в техническую теорию устойчивости движения* (Государственное издательство физико-математической литературы, Москва, 1962).
[5] Ляпунов, А. М., *Общая задача об устойчивости движения* (Гостехиздат, Москва, 1950).

[6] Мартынюк, А. А., *Техническая устойчивость в динамике* (Издательство Техника, Киев, 1973).
[7] Четаев, Н. Г., «О некоторых вопросах, относящихся к задаче об устойчивости неустановившихся движений», *Прикладная математика и механика*, **24** (1960) 6—19.

(Beérkezett: 1977. julius 1.)

KLINCSIK MIHÁLY
JÓZSEF ATTILA TUDOMÁNYEGYETEM BOLYAI INTÉZET
6720 SZEGED, ARADI VÉRTANÚK TERE 1.

ИССЛЕДОВАНИЯ УСТОЙЧИВОСТИ НА КОНЕЧНОМ ИНТЕРВАЛЕ ВРЕМЕНИ С ПОМОЩЬЮ ДИФФЕРЕНЦИАЛЬНЫХ НЕРАВЕНСТРВ

М. Клинчик

В настоящей статье дается достаточное условие устойчивости на конечном интервале времени с использованием второго метода Ляпунова и теории дифференциальных неравенств. Это условие обобщает результат Ван Дань-чжи и С. Я. Степанова [3]. С помощью этого условия исследуется устойчивость решений нелинейных дифференциальных уравнений на конечном интервале времени по первому приближению. Теоремы прилогаются к исследованию устойчивости движений некоторых механических систем.

Alkalmazott Matematikai Lapok 2 (1976) 289—294

AZ INVERZ ASSEMBLER EGY ÁLTALÁNOS MODELLJE

NAGY MIHÁLY ÉS VARGA LÁSZLÓ
Budapest

Az inverz assembler a program gépi kódú formájából állítja elő a program assembly nyelvű formáját, és a programok adaptálásának fontos eszköze. A dolgozatban definiáljuk a gépi kódú program, az assembly nyelvű program és az inverz assembler fogalmát. Megadjuk az általunk kidolgozott konkrét inverz assemblerek egy általános algoritmusát, amely a VDL gráf bejárásának stratégiáján alapszik. Definíciós eszközként a bécsi definíciós nyelvet (VDL-t) használjuk.

1. Bevezetés

Ma már hatalmas mennyiségű szellemi érték halmozódott fel programok formájában a számítóközpontok könyvtáraiban. Komoly népgazdasági érdekek fűződnek a meglevő programok hasznosításához. A kipróbált programoknak gyakran csak a gépikódú formájához jutunk hozzá. Adaptációnál felmerül annak a szükségessége, hogy a program bizonyos tulajdonságait mélyebben megértsük, mint azok a felhasználói leírásból megállapíthatók. Esetleg felmerül a program módosításának szükségessége is. Ilyenkor nélkülözhetetlen programeszköz az inverz assembler, amely a gépikódú forma alapján előállítja a program assembly nyelvű formáját.

Az *MTA Központi Fizikai Kutató Intézet*ben eddig több inverz assemblert dolgoztunk ki, ezek felhasználói leírásai megtalálhatók a *KFKI Programkönyvtár*ában. A következőkben a konkrét inverz assemblerek általános algoritmusát fogalmazzuk meg bécsi definíciós nyelven (VDL-ben). A VDL leírása megtalálható például a [1], [2] munkákban. A modell alapját a VDL-gráf és annak bejárási algoritmusa képezi, amelyek a [3], [4] munkákban találhatók meg.

2. A probléma megfogalmazása

A gépikódú program a kódok egy rendezett halmaza. A kód funkciója szerint lehet utasítás vagy adat. Mi most olyan programokkal kívánunk foglalkozni csak, amelyekben az utasításkódok és az adatkódok vegyesen helyezkednek el.

Az utasításkódról feltesszük, hogy azt a program adatként nem használja, azaz utasításmódosítást nem hajt végre.

Tegyük fel, hogy a program utasításkódjai mind explicit módon tartalmazzák a végrehajtás során közvetlenül utánuk következő utasításkód címét. Ez egyrészt azt jelenti, hogy olyan programokkal foglalkozunk, amelyekben nincs számított vezérlésátadó utasításkód, másrészt a nem vezérlésátadó utasításkódokhoz is hozzárendelve képzeljük el a következő utasításkód címét.

Ez az utóbbi feltételezés nem jelent a gyakorlati modellekhez képest lényeges megszorítást, hiszen az utasítás kódjához és egy utasításszámlálóhoz mindig hozzárendelhető az az algoritmus, amely a következő utasítás címét szolgáltatja.

A számított vezérlésátadás viszont több gép utasításkészletében is megtalálható. A 3. pontban röviden vázolni fogjuk azt a gyakorlati módszert, amellyel a megadott algoritmus ilyenkor is alkalmassá tehető a feladat megoldására.

Nevezzük a program utasításainak halmazát *tényleges programnak*. Tegyük fel, hogy a tényleges programnak véges sok indítási pontja van, és a programnak nincsenek „felesleges" utasításai, azaz bármely utasításhoz tartozik olyan indítási pont, amelyből elindítva a programot, az utasítás végrehajtásra kerül. A program utasításai tehát az indítási pontokból kiindulva elérhetők, a program bejárható. Ezek alapján a gépi kódú program absztrakt fogalmát a következőképpen definiálhatjuk:

2.1. DEFINÍCIÓ. A gépi kódú programok halmaza:

$$\{p \mid \text{is-code-list } (p)\}$$

ahol

$$\text{is-code} = \text{is-data} \vee \text{is-stmt}.$$

A tényleges programok halmaza:

$$\{t \mid \text{is-instr-graph } (t)\},$$

részletesen

$$\text{is-instr-graph} = (\{\langle s\text{: is-stmt}\rangle \mid \text{is-select } (s)\})$$

$$\text{is-stmt} = (\langle s\text{-value: is-instr}\rangle,$$

$$\langle s\text{-desc: is-select-list}\rangle),$$

ahol a gráf [4]-ben megadott formális definíciójában most az is-node helyére az is-stmt predikátumot írtuk.

2.2. DEFINÍCIÓ. Legyen

$$\text{is-code-list } (p) = T$$

és

$$\text{is-instr-graph } (t) = T.$$

A t tényleges programot akkor és csak akkor nevezzük a p program részének, ha

$$(\forall n \in t)(\exists i)(\text{elem } (i)(p) = n).$$

2.3. DEFINÍCIÓ. Legyen

$$\text{address } (s)$$

az a függvény, amely adott p program és annak részét képező t tényleges program esetén a következő kölcsönös és egyértelmű leképezést valósítja meg:

$$(\forall s(t) \in t)(\text{elem } (\text{address } (s))(p) = s(t)).$$

2.4. DEFINÍCIÓ. Legyen

$$\{a \mid \text{is-assembly } (a)\}$$

az assembly nyelvű utasítások halmaza. Legyen

$$\text{assembler}\ (x)$$

az a függvény, amely a

$$\{n|\text{is-stmt}\ (n)\}$$

halmazt egyértelműen leképezi a

$$\{a|\text{is-assembly}\ (a)\}$$

halmazra.

2.5. DEFINÍCIÓ. Legyen

$$\text{is-code-list}(p) = T$$

és t a p program tényleges programrésze. Az inverz assemblert a következő leképezéssel definiáljuk:

$$p' = \mu(p; \{\langle \text{elem}\ (\text{address}\ (s)): \text{assembler}\ (s(t))\rangle | s(t) \in t\}).$$

3. Az általános inverz assembler algoritmusa

3.1. DEFINÍCIÓ. Legyen az inverz assembler feladatát megvalósító absztrakt gép állapotainak halmaza:

$$\{\xi|\text{is-state}\ (\xi)\},$$

ahol

$$\begin{aligned}\text{is-state} = (&\langle s\text{-input: is-code-list}\rangle,\\ &\langle s\text{-table: is-value-set}\rangle,\\ &\langle s\text{-output: is-assembly-code-list}\rangle,\\ &\langle s\text{-control: is-control}\rangle),\end{aligned}$$

és itt

$$\text{is-value} = \{T, F\}$$

$$\text{is-assembly-code} = \text{is-data} \vee \text{is-assembly}.$$

3.2. DEFINÍCIÓ. Legyen az absztrakt gép kezdeti állapota:

$$\begin{aligned}\xi_0 = \mu_0(&\langle s\text{-input: } p\rangle,\\ &\langle s\text{-table: } t_0\rangle,\\ &\langle s\text{-output: } \langle\rangle\rangle,\\ &\langle s\text{-control: } \textbf{inverz-assembler}\ (p)\rangle),\end{aligned}$$

ahol p program, amelynek tényleges programrésze t és

$$t_0 = \mu_0(\{\langle s: F\rangle | \text{is-master}\ (s(t))\})$$

és itt

$$\{n|\text{is-master}\ (n)\}$$

a t indítási pontjainak halmaza (lásd [4]).

Legyen *next-selector* (x) az a függvény, amely az x táblázatot egy olyan s szelektorra képezi le, amelyre $s(x)=F$, ha ilyen szelektor létezik, és különben eredményül az Ω objektumot adja.

3.1. TÉTEL. Legyen ξ_0 a 3.2 definícióban megadott, akkor a következő program megvalósítja az inverz assembler feladatát:

inverz-assembler $(p)=$

next-selector (*s*-table $(\xi)) = \Omega \rightarrow$ **translate** (p, i);

i: **pass** (1)

$T \rightarrow$ **inverz-assembler** (p);

process-stmt (n, s);

n: **next-stmt** (s);

s: **next-address**

pass $(t) =$

PASS: t

next-address $=$

PASS: *next-selector* (*s*-table (ξ))

next-stmt $(s) =$

PASS: elem (address (s))(*s*-input (ξ))

process-stmt $(n, s) =$

process-address (s),

process-desc (*s-desc* (n)),

process-desc $(w) =$

length $(w) = 0 \rightarrow$ **null**

$T \rightarrow$ **process-desc** (tail (w));

set (head (w))

set $(s) =$

s(*s*-table $(\xi)) = \Omega \rightarrow$ **link** (s)

$T \rightarrow$ **null**

link $(s) =$

s-table: μ(*s*-table; $\langle s: F\rangle$)

process-address $(s) =$

s-table: μ(*s*-table; $\langle s: T\rangle$)

translate $(p, i) =$

length $(p) = \Omega \rightarrow$ **null**

$T \rightarrow$ **translate** (tail (p), $i+1$);

trans (head (p), i)

trans $(n, i) =$

address$^{-1}(i)$(s-table (ξ)) $= \Omega \rightarrow$ **insert** (n, i)

$T \rightarrow$ **insert** (assembler (n), i)

insert $(v, i) =$

s-output: μ(s-output (ξ); $\langle$elem (i): $v\rangle$),

ahol

address$^{-1}(i)$

az address (x) függvény inverzét jelöli.

Bizonyítás. A **inverz-assembler** (p) makroutasítás t gráf bejárási algoritmusát definiálja (lásd [4] 3.1 tétel). Ezért

inverz-assembler (s-input (ξ_0))

program végrehajtása után

$$(\forall s, s(t) \neq \Omega)(s(s\text{-table } (\xi)) = T)$$

és

$$(\neg\exists s)(s(\text{-table } (\xi)) = F).$$

A

translate (s-input (ξ_0), 1)

program pedig a $p = s-\text{input } (\xi_0)$ program minden kódjára alkalmazza a *trans* műveletet. A

trans(kód$_i$, i)

művelet viszont nyilvánvalóan a 2.5 definícióban megadott módon állítja elő a

$$p' = s\text{-output } (\xi)$$

programot.

A fenti algoritmus szerint készítettük el a *Központi Fizikai Kutató Intézetben* azt az inverz assemblert, amely a TPA—1140-es gép gépi kódú programjait fordítja át assembly nyelvű formára.

A modell gyakorlati szempontból fontos hiányossága, hogy nem oldja meg a számított vezérlésátadások hivatkozásainak problémáját. Ez a következőképpen oldható meg:

A programot alkalmasan választott kezdőadatok mellett lefuttatjuk. Egy nyomkövető programmal megállapítjuk azoknak az utasításoknak a címeit, amelyek a program lefuttatása során végrehajtásra kerültek. Ezután az így kapott címeket mind belépési pontoknak tekintve, alkalmazzuk a fent ismertetett algoritmust.

Ekkor a program végrehajtása során bejárt útból a feltételes vezérlésátadásoknál leágazó utakat is nagy valószínűséggel be fogjuk járni.

A program lefuttatása történhet értelmező program segítségével is. Mi a TPA—1140-es gép esetében ezt a megoldást választottuk.

IRODALOM

[1] WEGNER, P., "The Vienna definition language", *Computing Surveys* **4** (1972).
[2] LEE, A. N., *Computer Semantics* (Van-Nostrand Reinhold Co., 1972).
[3] VARGA, L., „Adatszerkezetek absztrakt szintaxisa és szemantikája", *Alkalmazott Matematikai Lapok* **2** (1976) 41—55.
[4] VARGA, L., "The VDL graph", *KFKI*—76—28.

(Beérkezett: 1976. október 11.)

NAGY MIHÁLY ÉS VARGA LÁSZLÓ
MTA KÖZPONTI FIZIKAI KUTATÓ INTÉZET
1525 BUDAPEST, 114. POSTAFIÓK 49.

A GENERAL MODEL OF INVERSE ASSEMBLER

M. NAGY and L. VARGA

The inverse assembler translates programs in machine code form into assembly form. It is an important tool of program adaptation. In this paper the notions of machine code program, assembly program and inverse assembler are defined. The mathematical model of certain concrete inverse assemblers is given on the basis of the VDL graph walking algorithm. The vehicle for the definition of the model is the *Vienna Definition Language* (VDL).

Alkalmazott Matematkai Lapok 2 *(1976) 295—306*

AZ ICL SYSTEM 4/70 „MULTIJOB SUPERVISOR" HATÉKONYSÁGI ÉRTÉKELÉSE

ASZTALOS DOMONKOS
Budapest

A cikk célja, hogy ismertesse az ICL System 4/70 számítógépen üzemelő *Multijob operációs rendszer* erőforrás igényeit, különös tekintettel a processzor idő felhasználás megoszlására a különböző rutinok között. A szerző mérési adatok alapján kimutatja, hogy a multiprogramozású, időosztásos rendszer központi vezérlőprogramjában, a *Supervisor*ban, a folyamatok szinkronizált végrehajtását vezérlő rutinok közül néhány hatékonysági szempontból nem megfelelő. Viszonylag kevés módosítással jelentős processzor-idő takarítható meg, az adott felhasználásnak kb. 9—11%-a.

1. Bevezetés

Az utóbbi években nagy lépésekkel haladt előre az operációs rendszerek elmélete, egyre több cikk és monográfia jelenik meg az operációs rendszerek logikai tervezéséről [3, 4]. Az elsődlegesen megoldandó probléma operációs rendszerek tervezése során az erőforrások megfelelő szétosztása a rendszerben élő folyamatok között. A probléma technikai megoldását a folyamatok szinkronizált végrehajtása adja. Nagyobb időosztásos rendszerekben az ehhez szükséges adminisztrációs idő (CPU-idő) jelentős hányadát teszi ki az összes időnek. Részben ezért szükségessé vált, hogy a bonyolult logikájú rendszerek hatékonysági szempontok alapján kiértékelésre kerüljenek. A kiértékelés egyik alapvető fázisa a mérés. Sajnos, jelenleg a szakirodalomban még kevés cikk található, amely mérési eredményekről számol be. Az ICL System 4/70 *Multijob rendszer*ével többékevésbé összevethető időosztásos rendszerekről közölnek mérési adatokat a [5, 6, 7, 8] cikkek.

A CPU-idő felhasználásának megoszlásáról — amely jelen cikk fő tárgya —, a [6] irodalomban található megfelelően részletes ismertetés. Egyébként három állapot tartamával szokták jellemezni a CPU-idő megoszlását. Ezek: *supervisor, felhasználói és nyugalmi állapot.* Az ismertetett rendszerek mindegyikénél a *supervisor* állapotban felhasznált CPU-idő az összidőnek nem kevesebb, mint 30—40%-át tette ki. Ez az egy adat már önmagában indokolja, hogy részletesebb vizsgálat tárgya legyen az operációs rendszerek adminisztrációs idejének megoszlása a különböző szolgáltatási igények szerint.

Az ICL System 4/70 számítógépen az időosztásos üzemmódot a *Multijob operációs rendszer* biztosítja. Az operációs rendszer magja a *Supervisor*, amelynek alapvető feladata a multiprogramozás vezérlése, az átviteli igények adminisztrációja, beleértve a telekommunikációt is, és a file-okhoz való hozzáférés szabályozása. Ebben a cikkben ismertetésre kerülnek a *Supervisor* erőforrás igényének vizsgálata során kapott eredmények.

2. A Multijob Supervisor főbb jellemzői

Az ICL System 4/70 számítógép négy külön állapotban működhet, ezek: *P*1, *P*2, *P*3, *P*4. A felhasználói programok *P*1 állapotban kerülnek végrehajtásra, a *Supervisor* a *P*2 és *P*3 privilégizált állapotokban működik. *P*4 állapotba csak néhány súlyos *hardware* hiba esetén kerül a gép, diagnosztikai rutinok végrehajtása céljából.

A fentieknek megfelelően a *Supervisor*ral kapcsolatosan *P*3- és *P*2-kódról beszélünk, amely a végrehajtás állapotának felel meg. A *Supervisor* rutinjainak egy része állandóan a memóriában található, a többi csak igény esetén kerül betöltésre az erre kijelölt memóriaterületre. A *P*3-kód a magasabb prioritású, teljes egészében állandóan a memóriában van és végrehajtása során nem megszakítható. Feladata a megszakítások elemzése, a magas prioritású igények azonnali végrehajtása (pl. *I/O igény inicializálása vagy terminálása, P*2 *állapotú Supervisor rutinok aktivizálása, rutinok össze- és szétkapcsolása*), kimeneteli ágon a CPU ütemezése. A *P*2-kód egy része szintén rezidens, amit a végrehajtás gyakorisága és a rendszer logikája indokol, míg a rutinok nagyobb része csak igény esetén kerül be a memóriába.

A *Supervisor P*3-kód része csak megszakításkor kapja meg a vezérlést. A normál üzemmód során megszakítást az alábbi események okoznak:

a) átvitel befejeződése,
b) *Supervisor Call* (SVC) utasítás végrehajtása,
c) óramegszakítás kb. 0,5 másodpercenként.

Ennek megfelelően egy aktív *P*1 vagy *P*2 rutin csak SVC utasítás végrehajtása útján kérheti a *Supervisor* szolgáltatásait. Több *P*2 rutin kapcsolódhat össze egy hívásláncban.

A *P*3-kód a már korábban röviden említett feladatoknak megfelelően az alábbi fontosabb rutinokra osztható:

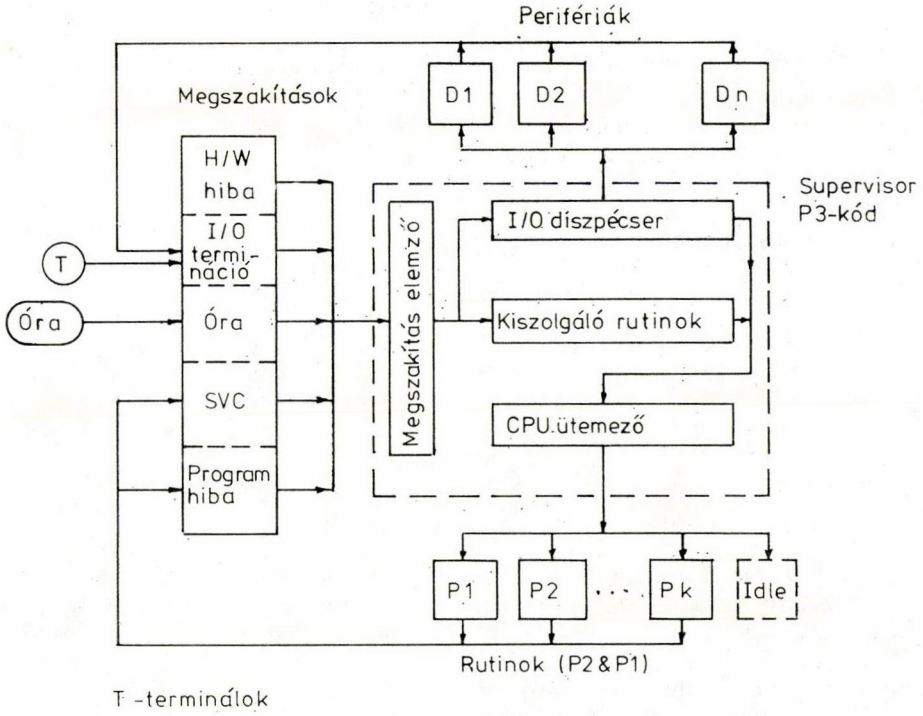
a) megszakítás elemzése,
b) rutinok összekapcsolása,
c) rutinok szétkapcsolása,
d) CPU ütemezése,
e) *V*- és *P*-műveletet realizáló rutin (kizárólagos erőforrások kezelése, ld. [1], — VOP/POP rutin),
f) logikai várakoztatás — felszabadítás, — WAIT/UNWAIT rutin,
g) átvitel kezdeményezése,
h) átvitel terminációja.

A *P*3-kód kapcsolatát a rendszer többi elemével az 1. ábra szemlélteti.

A *P*3-kódból a kilépés a CPU ütemezőn keresztül történik, amelynek feladata a legmagasabb prioritású nem várakozó *P*2 vagy *P*1 rutin kiválasztása végrehajtásra. Ha ilyen nincs, a gép *P*2-nyugalmi állapotba (*idle*) kerül a következő megszakításig. Ez az eset áll fenn pl. mikor minden aktív rutin átvitel befejeződésére vagy az operátor beavatkozására vár.

A prioritás megoszlása csökkenő sorrendben:

a) *P*2 *Rezidens rutinok*
b) *P*2 *Tranziens rutinok*
c) *P*1 *rutinok* (felhasználói programok)



1. ábra. A Supervisor *P*3-kód helye a rendszerben.

A *P*2 *Rezidens rutinok* egymáshoz viszonyított prioritása rögzített. A *P*2 *Tranziens rutinok* prioritását a memóriába való betöltés sorrendje határozza meg. A *P*1 *rutinok* prioritása kívülről adható meg, egymáshoz viszonyított értékük végrehajtás során csak operátori beavatkozással változtatható.

A *P*2 *Tranziens rutinok* és a működéshez szükséges rendszer blokkok a *Dinamikus Pufferterület*en kerülnek elhelyezésre. Ez utóbbi nagyságától (pufferek száma) jelentős mértékben függ a rendszer válaszideje.

A legfontosabb *P*2 rutinok:

a) Dinamikus Pufferterület kezelés,

b) Katalógizált file-kezelés rutinjai,

c) Telekommunikáció vezérlése, terminál parancsok értelmezése, végrehajtása,

d) Időosztásos munkák kezelése,

e) Operátor parancsok értelmezése, végrehajtásuk kezdeményezése.

3. A környezet leírása

Az OTSZK ICL System 4/70 számítógép konfigurációja:
448 kb. operatív tár,
3×EDS60 60 Mb kapacitású lemez meghajtó,
6×RDS 7 Mb kapacitású lemez meghajtó,
7×M9,
1×M7,
Szokásos lassú perifériák,
10 teletype, 2 VDU, 2 RDT *(Remote Data Terminal)*.

Az operatív tár felosztása:

Read-Only Memória	4 Kb
Supervisor P-3kód+táblák	19 Kb
Supervisor rez. *P*2-kód	75 Kb
Dinamikus Pufferterület	84 Kb
RIRO-*memória*	60 Kb
Batch-memória	206 Kb
	448 Kb

A RIRO-memória egyidőben csak egy programot tartalmazhat (nem multiprogramozható), de a program csak korlátozott ideig tartózkodhat a memóriában (az időszelet nagyságától függően), utána egy másik program gördül be a helyére. A RIRO-memóriához tartozik egy lemez-file, amely annyi 60 Kb-os partícióból áll, ahány program futhat időosztásos módon a RIRO-memóriában.

Jellege szerint a terhelés programfejlesztői tevékenység, a multi-programozás átlagos szintje (csak a felhasználói programokra vonatkoztatva): 1.8, az általuk felhasznált CPU-idő átlagosan a folyóidő 17—20%-a, míg az aktív telekommunikációs egységek száma 9.

A terminál tevékenység zömében file-manipulációból (létrehozás, módosítás, törlés, másolás) és programok (elsősorban fordítók) indításából áll. Viszonylag ritka, hogy a felhasználó egy futó programmal kommunikál. A terminál tevékenység alacsony CPU-idő igényű, de nagy *file-aktivitás*sal jár.

4. A mérések lebonyolítása

A mérés eszköze egy *software monitor* (ld. [2]), amely minden megszakítás fellépésekor megkapja a vezérlést. Kezelése egyszerű, normál felhasználói programként futtatható. A *software monitor* azon része, mely a mérést végrehajtó kódot tartalmazza, könnyen változtatható, ezért mindig csak az igényeket kielégítő kódot kell beépíteni, ami révén csökken a monitor hely és idő igénye. A mérések során használt két monitor mindegyike 4 Kb memóriát köt le (ez a legkisebb leköthető terület), CPU igénye megszakításonként 63 μsec. Az összes CPU terhelés megoszlásának függvényében a monitor a CPU idő 3—5%-t használta fel.

Mindkét monitorral mért mennyiségek:
a) mérés időtartama,
b) megszakítások száma.

A MON—1 által mért mennyiségek:

a) CPU-idő megoszlása:

P3-idő,
P2-idő,
P1-idő,
RIRO-*idő*,
Batch-idő,
Monitor-idő,

b) A *Supervisor P2* rutinjainak használatával kapcsolatosan az alábbi mennyiségek:

aktivizálások száma,
felhasznált CPU-idő,
átvitelek száma.

A MON—2 által mért mennyiségek:

a) A megszakítások típus szerinti eloszlása, típusonként a felhasznált CPU-idő (összesen és az egy megszakításra vonatkoztatott átlag),

b) A *Supervisor P3*-kód szolgáltatásait igénylő SVC utasítások megoszlása típus és CPU-idő felhasználás szerint.

A két monitor nem egyidőben lett kifejlesztve, az első (MON—1) eredményei vezettek a második létrehozásához.

5. A MON—1 monitor eredményei

A CPU-idő felhasználásával kapcsolatosan ld. az 1. táblázatot. Ez hét mérés eredményét tartalmazza, amelynek összes időtartama 9212 mp (2.55 óra) volt.

Az átlagos CPU kihasználtság a hét mérésre kivetítve 56.8 %-os volt. Az 1. táblázat 2. és 3. sorát összehasonlítva megállapíthatjuk, hogy a *Supervisor* CPU igénye minden esetben meghaladta a felhasználói programokét. A 2. és 12—14. sorok tanulmányozásából kitűnik, hogy egy megszakítás kiszolgálásához szükséges *Supervisor* CPU-idő nagysága stabil értéket mutat, átlagosan 0,987 ms. Ez azt jelenti, hogy ha a *Supervisor*t a *hardware* kiegészítő részének tekintjük (ami teljesen jogos egy felhasználói program szempontjából), akkor a rendszer potenciálisan kb. 1000 megszakítás kiszolgálására képes másodpercenként. Természetesen, ez nem fordulhat elő, mert egy bizonyos terhelés mellett annyira megnő a rendszer válaszideje, hogy a terhelés növelése már nem válik lehetségessé.

Az 1. táblázat 13. és 14. sorai szerint az egy megszakításra jutó *P3* CPU idő jóval meghaladja az egy megszakításra jutó *P2* CPU idő értékét, az első kb. háromszorosa a másodiknak.

Ennek alapján a cikkben elhagyjuk a MON—1 monitor által mért, b) pontban felsorolt mennyiségek ismertetését, és a továbbiakban a *P3*-kód CPU-idő igényével foglalkozunk.

1. TÁBLÁZAT

CPU idő felhasználás megoszlása (MON—1)

	I.	II.	III.	IV.	V.	VI.	VII.
1. CPU/T(%)	69,90	62,7	25,70	59,70	58,80	42,10	59,90
2. SVCPU/T(%)	38,34	44,91	21,80	39,70	45,64	32,44	32,14
3. UCPU/T(%)	26,23	11,86	0,68	15,48	8,10	5,35	23,87
4. SVCPU/CPU(%)	54,8	71,6	84,7	66,4	77,4	76,9	53,6
5. P3CPU/CPU(%)	45,4	54,7	61,5	49,0	59,2	56,8	40,2
6. P2CPU/CPU(%)	9,1	16,9	23,2	17,4	18,1	20,1	13,4
7. P1CPU/CPU(%)	41,8	23,6	9,9	29,4	17,7	18,0	42,6
8. FCPU/CPU	4,4	4,6	6,6	3,4	3,9	5,0	2,7
9. RCPU/CPU	17,5	5,8	1,9	15,8	8,5	11,6	20,4
10. BCPU/CPU	19,9	13,1	1,3	10,1	5,2	1,1	19,5
11. MCPU/CPU	3,1	4,7	5,2	4,2	4,8	5,0	3,7
12. Megszakítás/sec	365	458	225	396	445	341	342
13. SVCPU/MSz (ms)	1,050	0,979	0,969	1,002	0,950	0,939	1,024
14. P3CPU/MSz (ms)	0,878	0,750	0,704	0,741	0,706	0,704	0,748
15. Mérés tartama (s)	545	1407	587	1879	1308	897	2589

T — mérés tartama
CPU — CPU idő összesen
SVCPU — Supervisor CPU-ideje
P3CPU — P3-kód CPU ideje
P2CPU — P2-kód CPU ideje
UCPU — felhasználói programok CPU ideje (UCPU = RCPU + BCPU)
P1CPU — P1-kód CPU ideje
FCPU — rendszer programok (Scheduler, Job Input, Print) CPU ideje
RCPU — RIRO programok CPU ideje
BCPU — Batch programok CPU ideje
MCPU — Monitor CPU ideje
MSz — megszakítás

6. A MON—2 monitor eredményei

Az 1. táblázat 5—6. és 13—14. sorából kitűnik, hogy a *Supervisor P*3 kód lényegesen nagyobb CPU igénnyel bír, mint a *P*2-kód. Ezért szükségesnek tartottam, hogy a *P*3 rutinok CPU igényéről, aktivizálásuk gyakoriságáról részletes képet kapjunk mérések útján.

A *P*3-kódnak, mivel nem megszakítható, csak CPU igénye van. *P*3-kódba belépni csak megszakítás útján lehet, ezért a mérések során a megszakítások típus szerint kerültek számlálásra. A 2. táblázat a globális mutatókat foglalja össze, amelyeket össze lehet hasonlítani az 1. táblázat megfelelő soraival. Látható, hogy a terhelés mértékét az egy másodpercre jutó megszakítások száma határozza meg. A 3. táblázat a megszakítások típus szerinti gyakoriságát és százalékos CPU felhasználását mutatja az összes *P*3 CPU felhasználásra vonatkoztatva. A típusok közül kimaradt az időmegszakítás, amely kb. fél másodpercenként lép fel és az eltelt idővel egyenesen arányos terhelést jelent.

A táblázatból látható, hogy a *multiplexor csatornáról* (lassú perifériák és terminálok) érkező megszakítások gyakorisága viszonylag szűk határok között mozog

2. TÁBLÁZAT

Összefoglaló adatok (MON2)

Mérés	I.	II.	III.	IV.	V.
Tartam	4955	2222	1207	2079	2628
P3CPU/T	23,9	32,5	18,3	37,0	26,5
MSz/sec	301	418	226	460	340
P3CPU/Msz	0,760	0,778	0,808	0,806	0,780

3. TÁBLÁZAT

Megszakítások megoszlása gyakoriság és CPU igény szerint

Mérés	I.		II.		III.		IV.		V.	
	gyak.	CPU	gyak.	CPU	gyak.	CPU	gyak.	CPU	gyak.	CPU
SVC	227	72,29	316	70,22	170	73,66	325	67,79	254	70,46
Diszk	37,7	17,95	51,0	17,67	22,3	13,79	47,7	12,18	44,8	19,45
Multiplexor	33,1	8,27	34,4	6,93	31,8	11,09	28,7	4,41	31,7	6,43
MT	1,3	0,54	14,2	4,57	0,19	0,09	56,4	15,19	7,30	2,89

(28,7—34,4/sec), míg a más típusú megszakítások gyakorisága a terhelés mértékétől és jellegétől függően nagyon eltérő értékeket vehet fel. A legnagyobb CPU igénye az SVC megszakításoknak van, ezért ezekkel célszerű részletesebben foglalkozni. SVC utasítást egy *P*1 vagy *P*2 rutin adhat ki, és az utasítás rendeltetése szerint az alábbi kategóriák valamelyikébe tartozhat:

a) rutinok szinkronizált végrehajtása,

b) átviteli igény kezdeményezése.

A multiprogramozást a központi egység és az átviteli egységek autonóm működése teszi lehetővé, és lényege, hogy egyazon időben több aktív folyamat vagy rutin létezhet a rendszerben. Általában egy aktív folyamat kétféle állapotban lehet: vagy vár valamilyen erőforrásra vagy szolgáltatást kap valamelyik erőforrástól. A *hardware*-szintű erőforrásoknak három fő típusáról beszélhetünk: operatív tár, központi egység, átviteli egység. Az operatív tárban egyszerre több rutin is tartózkodhat, míg a központi egység vagy egy átviteli egység mindig csak egy igényt (rutint) tud kiszolgálni. Ennek alapján a két utóbbi erőforrást kizárólagosnak nevezzük. A *hardware* erőforrásokon kívül léteznek *software* erőforrások (rutinok, *file*-ok vagy táblázatok), amelyek között szintén vannak kizárólagosak (pl. *File Katalógus*).

A rutinok szinkronizált végrehajtása terén a *P*2-rutinok vezérlése a bonyolultabb probléma, a *Multijob* rendszerben csak ezek esetében van szükség kizárólagos *software* erőforrások használatára. A *Supervisor* főbb feladatai ezen a területen:

1. rutinok össze- és szétkapcsolása *(setup, return, wait/unwait)*,
2. egyéb kizárólagos *software* erőforrások kezelése (VOP/POP),
3. átvitel befejeződésére való várakoztatás (*wait* I/O),
4. CPU ütemezés.

4. TÁBLÁZAT

SVC megszakítások megoszlása gyakoriság és CPU igény szerint

Mérés	I.		II.		III.		IV.		V.	
	gyak.	CPU	gyak.	CPU	gyak.	CPU	gyak.	CPU	gyak.	CPU
Return	66,9	24,31	77,8	18,08	53,7	32,33	74,3	13,41	66,7	20,68
Setup	32,8	5,84	43,0	5,71	25,1	5,23	39,6	4,24	30,1	5,12
VOP/POP	64,4	10,38	96,4	11,55	44,6	9,02	90,4	8,89	86,3	12,23
WAIT I/0	18,8	11,26	29,5	10,76	14,4	9,12	19,1	4,95	22,5	12,00
W/UW	12,3	5,08	16,8	4,84	9,0	4,63	13,7	3,34	12,0	4,35
Összes		56,87		50,94		60,33		34,83		54,38

A 4. táblázat tartalmazza a szinkronizált végrehajtás adminisztrációjának CPU igényét, az egyes *P*3 rutinok aktivizálásának gyakoriságát. Az összesített CPU igény csak akkor volt a *P*3 CPU felhasználás 50%-a alatt, amikor a felhasználói programok átviteli igénye jelentősen megnőtt (ld. a IV. mérés adatait a 3. és 4. táblázatban). A továbbiakban a VOP/POP és Wait I/0 rutinokkal valamint a CPU ütemezővel foglalkozom részletesebben.

VOP/POP rutin

A VOP/POP *P*3 rutin feladata a nem kód jellegű (táblázat és file) kizárólagos software erőforrások használatának szinkronizálása. A rutin neve a *V* és *P* művelet (V-OPeration, P-OPeration) angol elnevezésének rövidítéséből származik. VOP/POP SVC utasítást csak a *P*2 rutinok adhatnak ki. VOP műveletre van szükség a kritikus szekcióba való belépéskor és POP műveletre az onnan való kilépéskor. A *Multijob* rendszerben 14 erőforrásra adható ki *V* és *P*-művelet, úm. *File Katalógus*, *Job-sor*, *láncolt blokkok*, *szabad sávok file*-ja, stb. Mindegyik erőforráshoz tartozik egy számláló, amelynek kezdeti értéke nulla. *V*-művelet esetén a számláló értéke 1-gyel megnő és ha értéke nagyobb egynél, akkor a VOP/POP rutin a hívó rutint felfüggeszti. *P*-művelet esetén, ha a számláló értéke nagyobb nullánál, akkor a számláló értéke eggyel csökken. Ha ezután értéke nagyobb nullánál, akkor a VOP/POP rutin felszabadítja az adott erőforrásra várakozó rutinok közül a legnagyobb prioritással bírót.

A HOARE által tervezett monitor fogalmat [9] felhasználva a fenti VOP/POP műveletnek megfelelő monitor:

```
jobqueue: monitor
sem: integer
nonbusy: condition
procedure vop
begin sem = sem+1;
if sem>1 then nonbusy.wait
end;
```

```
procedure pop
begin
if sem > 1 then nonbusy.signal
if sem > 0 then sem = sem − 1
end;
sem = 0
end jobqueue
```

Az eljárások hívása a

```
jobqueue.vop
jobqueue.pop
```

utasításokkal történik. A két eljárás egymást kizáró végrehajtását a *P*3-kódban való realizáció biztosítja. Ennek hátránya, hogy a CPU ütemező végrehajtására (ld. 1. ábra) csak a *wait* vagy *signal* műveleteknél van szükség. További vizsgálatok megmutatták, hogy a hívásoknak átlagosan 94%-ában nincs szükség a *wait* vagy *signal* és ezzel együtt a CPU ütemező végrehajtására. Ily módon, ha a vop és pop kizárólagos végrehajtásnak más eszköze is van, akkor azt felesleges *P*3 állapotban megtenni. A SYSTEM 4/70 gépen ez az eszköz az LSP *(Load Scratchpad)* utasítás, amellyel az adott esetben a *P*2 állapot *Interrupt Mask Register* értéke változtatható úgy, hogy a megszakítások először le legyenek tiltva, majd ismételt végrehajtás után újra érvényesülhessenek. Megszakítás generálására csak a *wait* vagy *signal* igénye esetén van szükség. A jelenlegi rendszerben ezzel a változtatással a *P*3 állapotban felhasznált CPU idő 7—8%-a lenne megtakarítható.

A fenti példa jó illusztrációja egyrészt annak, hogy az elméleti eredmények gyakorlati alkalmazása még sok részlet elemzését teszi szükségessé annak megértésén túl, hogy az elmélet eredményeit hol lehet és kell alkalmazni, másrészt annak, hogy a rendszer bizonyos kritikus részeinek megítélésekor a hatékonyság majdnem olyan rangú minősítő tényező, mint a logikai ellentmondástól való mentesség.

Wait I/0 rutin

Több operációs rendszerben megengedett, hogy a program egy átviteli igény kiadása után is tovább folytathassa a foldolgozást. Ekkor azonban a programnak jeleznie kell, amikor már szüksége van az átvitel eredményére. Ha az átvitel még nem fejeződött be, akkor a programot a *Supervisor* felfüggeszti az átvitel végleges kiszolgálásáig. Tehát ebben az esetben is folyamatok szinkronizált végrehajtásának igényével találkozunk.

A fenti eljárás alkalmazásának hatékonysága csak statisztikailag igazolható. Két tapasztalati statisztikát kell mérni:

a) a programok felfüggesztéscinek száma,

b) az átvitel kezdeményezése és a jelzés között felhasznált CPU idő egy átvitelre átlagosan (t_{OVL}).

Egyszeri felfüggesztés átlagos *CPU igénye* (t_{CPU}) 1193 μs. A felfüggesztés tapasztalati valószínűsége az átvitelek számára vonatkoztatva kb. 0,9 (ld. 5. táblázat), vagyis t_{OVL} lényegesen kisebb mint egy átvitel kiszolgálási ideje. Az átvitelek kezelésének fenti módja akkor hatékony, ha $t_{OVL} \gg t_{CPU}$. t_{OVL} értéke nem lett mérve, de

5. TÁBLÁZAT

WAIT I/0 rutin használata

Mérés	I.	II.	III.	IV.	V.
1. Wait nélküli átvitelek gyakorisága	20,10	32,67	15,69	22,07	25,35
2. WAIT I/0 gyakorisága	18,8	29,5	14,4	19,1	22,5
2./1.	0,935	0,904	0,923	0,868	0,889

a *Supervisor* kódjának elemzése alapján megállapítható, hogy a *Supervisor* átviteli igényei esetén $t_{OVL} \simeq 0$. Gyakorlatilag csak a rendszer státuszú felhasználói programok használják ki a CPU és I/0 tevékenység átlapolásának lehetőségét egy programon belül. Ebből megállapítható, hogy az átvitelek inicializálása utáni állapotra vonatkozó egységes alapértelmezés a *Supervisor* esetén hatékonysági szempontból nem megfelelő. A *Supervisor* rutinjai részére az alapértelmezés megváltoztatása (azonnali várakozás inicializálás után) 3—4% CPU-idő megtakarítást jelentene *P*3 állapotban.

CPU-ütemezés

A *Supervisor* erőforrás igénye csökken, ha csökken az egy megszakítás kiszolgálásához szükséges átlagos CPU idő nagysága. Ha ebből a megközelítésből foglalkozunk a problémával, akkor először a *Supervisor* leggyakrabban aktivizált rutinjainak működését kell analizálni. Az 1. ábra szerint a két leggyakrabban aktivizált rutin a *Megszakítás elemző* és a CPU-ütemező, amelyek minden megszakításkor megkapják a vezérlést.

A megszakítás elemző rutin feladata a megszakítás típusának megállapítása és a vezérlést átadni a megfelelő *P*3 rutinnak. Ezt a feladatot a rutin a minimális CPU felhasználással látja el, itt tehát javítás nehezen képzelhető el.

Másként áll a helyzet a CPU-ütemezővel. Vannak olyan megszakítást okozó események, amelyek kiszolgálása után nyilvánvalóan ugyanaz a rutin kapja meg a vezérlést, amely a megszakítás előtt aktív volt. Ezt a tényt a jelenlegi CPU-ütemező csak annyiban használja ki, hogy nem hajtja végre feleslegesen a megszakított rutin regisztereinek mentését és újratöltését. Ugyanakkor az ütemező minden esetben végrehajtja a keresést, míg megtalálja a legmagasabb prioritású aktív, nem várakozó rutint.

A következőkben felsorolom azokat az eseményeket, amelyeknél a jelenlegi rendszerben felesleges a CPU-ütemezés. Ezek:

a) Átvitel befejeződése egy olyan rutin részére, amelynek prioritása alacsonyabb a megszakítás előtt aktív rutin prioritásánál (a megszakítás előtt nem nyugalmi állapotban volt a CPU),

b) NDFA-blokk vagy várakozás nélküli egyéb átvitel (EXCP, EXDP) inicializálása,

c) VOP megszakítás után, ha a megfelelő számláló értéke a megszakítás pillanatában nulla volt,

d) POP megszakítás után, ha a megfelelő számláló értéke a megszakítás pillanatában kettőnél kisebb volt.

A CPU-ütemezés szükségessége szempontjából tehát a megszakítást okozó események két osztályba sorolhatók. A hovatartozás jelölésére be lehetne vezetni egy ‚CPU-ütemezés' logikai változót, ami a CPU ütemező paramétere lehet. Ezzel a megoldással további jelentős CPU igény csökkenést lehetne elérni.

6. TÁBLÁZAT

Egy megszakítás CPU igénye típusonként) usec)

Mérés	I.	II.	III.	IV.	V.
SVC	726	719	793	773	733
VOP/POP	369	390	369	363	376
W/UW	941	934	934	901	961
WAIT I/0	1371	1183	1156	961	1412
Setup	403	430	376	396	450
Return	827	753	1096	665	820
Átvitelek					
EXCP	1035	1123	1069	1049	1129
EXCPW	1277	1425	1412	1607	1230
EXDP	1345	2017	1143	921	1963
EXDPW	1324	1439	1163	1190	1356
GETF	1096	1109	1089	1136	1156
GETX	914	948	948	928	961
GETB	1304	1217	1143	1318	1324
PUTF	1089	1109	1116	1116	1170
PUTX	1008	1062	1028	1069	1076
Diszk. term.	1109	1126	1138	946	1151
Multiplexor term.	571	652	632	564	538

Végül a 6. táblázat adatai illusztrálják, hogy típusonként az egy megszakításra jutó CPU igény milyen értéket vett fel. Jelenleg nem tudom megmagyarázni, hogy néhány megszakítás típusnál a mérések közötti nagy eltéréseknek mi az oka (pl. WAIT I/0, EXDP).

7. Összefoglalás

A felhasználói rutinok szempontjából a *hardware* és a *Supervisor* egységeként jelenlevő számítógép teljesítményét többféleképp lehet növelni, amelynek egyik újta a *Supervisor* erőforrás igényének csökkentése.

Ez utóbbinak két módja van: megszakítások gyakoriságának viszonylagos csökkentése és/vagy az egy megszakítás kiszolgálásához szükséges átlagos CPU igény csökkentése. A megszakítások gyakoriságának viszonylagos csökkentésére tett javaslatnak tekinthető a VOP/POP és WAIT I/O ruinokról elmondottak, míg a CPU ütemezésre vonatkozó rész az egy megszakítás átlagos CPU igényének csökkentésével kapcsolatos.

A fenti eredmények nem lettek volna elérhetők a *software monitor* használata nélkül, ami alátámasztja a mérések jelentőségét olyan kulcsfontosságú *software*-termékek kiértékelése, minősítése esetén, mint a *Multijob Supervisor*.

IRODALOM

[1] DIJKSTRA, E. W., "Hierarchical ordering of sequential processes", *Acta Informatica* **1** (1971) 115—138.
[2] ASZTALOS, D., „Az operációs rendszer hatékonyságának vizsgálata", *OTSzK Közlemények* **2** (1974).
[3] BRINCH—HANSEN, P., *Operating System Principles* (Prentice-Hall, Englewood Cliffs, New Yersey, 1973).
[4] SHAW, A. C., *The Logical Design of Operating Systems* (Pretice-Hall, Englewood Cliffs, New Yersey, 1974).
[5] SHELNESS, N. H., STEPHENS, P. D. and WHITFIELD, H., "The Edinburgh Multi-Access System Scheduling and Allocation Procedures in the Resident Supervisor", in: *Operating Systems, Proc. of an International Symposium*, Ed. E. Gelenbe and C. Kaiser (Springer Verlag, 1974) 293—310.
[6] ADAMS, J. C. and MILLARD, G. E., "Performance Measurement on the Edinburgh Multi-Access System", in: *Proc. of International Computing Symposium*, Ed. E. Gelenbe and D. Potier (North-Holland, 1975) 105—112.
[7] JALICS, P. J. and LYNCH, W. C., "Selected Measurements of the PDP—10 TOPS—10 Timesharing Operating System", *IFIP 74. Software vol.* 242—246.
[8] CHIU, W., DUMONT, D. and WOOD, R., "Performance Analysis of a Multiprogrammed Computer System", *IBM Journal of R&D* **19** (1975) 263—271.
[9] HOARE, C. A. R., Monitors: An Operating System Structuring Concept, *CACM* **17** (1974) 549—557.

(Beérkezett: 1976. szeptember 2.)

(Újra beérkezett: 1977. február 3.)

ASZTALOS DOMONKOS
OT SZÁMÍTÁSTECHNIKAI KÖZPONTJA
1149 BUDAPEST, XIV. ANGOL U. 27.

PERFORMANCE EVALUATION OF THE MULTIJOB OPERATING SYSTEM

D. ASZTALOS

The main purpose of the paper is to describe the distribution both of the request types to the *Supervisor* and the CPU time among the different parts of the *Supervisor* on a SYSTEM 4/70 Computer. The data to characterize the above distributions were collected by a *software monitor* [2]. Those data published in a few papers [5, 6, 7, 8] suggest that the 30—40% of CPU-time is spent in system-state and that was the reason to draw a more detailed picture of the CPU usage in the *Supervisor*.

One of the main results is that the service of one interrupt by the *Supervisor* requires an average CPU-time of 1 ms. On the base of this fact it is important to reveal the points in the *Supervisor* giving poor performance. The author believes he detected two such points: 1. the realization of the semaphore operations; 2. the WAIT I/0 routine. Some suggestions are made in both cases to improve the situation.

Some of the results may be particular, but the use of the measurement techniques described in the paper may improve our understanding of the modern complex operating systems.

Alkalmazott Matematikai Lapok **2** *(1976) 307—313*

BALRÓL FAKTORIZALT (LF) NYELVTANOK SZÜKSÉGES ÉS ELÉGSÉGES FELTÉTELÉRŐL

VU-LUC
Budapest

Az osztott nyelvtanokat mintegy tíz éve tanulmányozzák és az utóbbi években a fordítóprogramkészítés gyakorlatában jelentős szerepet kaptak [2]. Az osztott nyelvtanok különböző változatai közül az LF-nyelvtanok különösen egyszerű szintaktikus elemzést tesznek lehetővé, ugyanakkor alkalmazásuk általában a programozási nyelvek szintakszisának jelentős átalakítását követeli meg [3, 4]. Cikkünkben az LF-nyelvtanok eredeti definíciójának és az átalakító algoritmusokban felhasznált tulajdonságának ekvivalens voltát fogjuk bizonyítani.

1. Terminológia és jelölések

Az ábécé nem üres, véges halmaz, amelynek az elemeit szimbólumoknak nevezzük. A string valamely ábécé szimbólumainak véges sorozata, amelyet α, β, γ,...-val, a string hosszát $|\alpha|$-kel, az üres stringet ε-nal jelöljük.

String halmazokon a következő műveleteket definiáljuk. Két halmaz szorzata

$$AB = \{\alpha\beta | \alpha \in A \text{ és } \beta \in B\}.$$

Egy halmaz hatványa

$$V^n = \begin{cases} \{\varepsilon\}, & \text{ha} \quad n = 0 \\ VV^{n-1} & \text{különben.} \end{cases}$$

Egy halmaz lezárása

$$V^* = \bigcup_{0 \leqq i} V^i.$$

Környezetfüggetlen nyelvtannak — KFN — nevezünk egy $G=(N, T, S, P)$ négyest, ahol

— N a nemterminális szimbólumok ábécéje;
— T a terminális szimbólumok ábécéje, és
$N \cap T = \emptyset$; jelöljük V-vel N és T egyesítést,
$N \cup T = V$;
— S a kitüntetett szimbólum, amely N eleme;
— $P = \{X \to \alpha | X \in N, \alpha \in V^*\}$ a helyettesítési szabályok véges halmaza.

Cikkünkben a továbbiakban a G nyelvtanra az itt bevezetett betűjelöléseket fogjuk használni. A görög betűkkel V^* elemeit fogjuk jelölni.

Legyen G KFN. Azt mondjuk, hogy G-ben γ-ból közvetlenül levezethető δ, azaz $\gamma \Rightarrow \delta$, ha $\gamma = \alpha A \beta$, $\delta = \alpha\eta\beta$ és $A \to \eta \in P$. γ-ból levezethető δ, azaz $\gamma \Rightarrow + \delta$, ha létezik az $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_n$ $(n \geqq 1)$ stringeknek egy olyan sorozata, hogy $\gamma = \alpha_0 \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots$ $\ldots \Rightarrow \alpha_n = \delta$. Az $\langle \alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_n \rangle$-sorozatot δ-nak γ-ból való levezetésének nevezzük.

Azt mondjuk, hogy $\gamma \Rightarrow * \delta$, ha $\gamma = \delta$ vagy $\gamma \Rightarrow + \delta$. A fenti definícióhoz hasonlóan bevezetjük a baloldali levezetés fogalmát.

Azt mondjuk, hogy δ γ-ból balról közvetlenül levezethető, $\gamma \overset{L}{\Rightarrow} \delta$, ha $\gamma = xA\alpha \Rightarrow$ $\Rightarrow x\beta\alpha = \delta$, ahol $A \to \beta \in P$ és $x \in T^*$. δ balról levezethető γ-ból, $\gamma \overset{L}{\Rightarrow} + \delta$, ha létezik olyan $\langle \gamma = \alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_n = \delta \rangle$ levezetés ($n \geqq 1$), hogy $\alpha_{i-1} \overset{L}{\Rightarrow} \alpha_i$ ($1 \leqq i \leqq n$). Végül $\gamma \overset{L}{\Rightarrow} * \delta$, ha $\gamma = \delta$, vagy $\gamma \overset{L}{\Rightarrow} + \delta$.

Azt mondjuk, hogy egy G nyelvtan X nemterminális szimbóluma felesleges, ha nem létezik olyan levezetés, hogy $S \Rightarrow * wXy \Rightarrow * wxy$, ahol $w, x, y \in T^*$, azaz X nem jelenik meg egyetlen mondat levezetésében sem.

Egy nyelvtant redukáltnak nevezünk, ha nincsen benne felesleges szimbólum. A nyelvtan által generált nyelv az S-ből a P-beli szabályok felhasználásával levezethető terminális stringek halmaza,

$$L(G) = \{x \in T^* \mid S \Rightarrow * x\}.$$

A rövidség kedvéért az egy nemterminális szimbólumhoz tartozó szabályokat egybegyűjtve a szokásosnak megfelelően úgy írjuk, hogy

$$A \to \alpha_1 | \alpha_2 | \ldots | \alpha_l$$

és α_i-t ($1 \leqq i \leqq l$) az A nemterminális szimbólum i-edik alternatívájának nevezzük.

A G KFN A nemterminális szimbólumát rekurzívnak nevezzük, ha létezik $A \Rightarrow + \alpha A \beta$. Ha $\alpha = \varepsilon$, akkor bal-, ha $\beta = \varepsilon$, akkor jobb-, ha pedig $\alpha \neq \varepsilon$ és $\beta \neq \varepsilon$ akkor belső rekurzivitásról beszélünk. A nyelvtant rekurzívnak mondjuk, ha van benne rekurzív szimbólum.

2. LF-nyelvtan és LF-nyelv

Tekintsük egy G nyelvtan valamely $A \to \alpha_1 | \alpha_2 | \ldots | \alpha_l$ szabályát. Legyen $\#$ speciális jel, $\# \notin V$; jelöljük $T' = T \cup \{\#\}$.

2.1. Definíció.

Az A nemterminális szimbólum i-edik alternatívájának balterminális halmaza

$$[A, \alpha_i] = \{a \in T' \mid S \# \Rightarrow * \alpha A \beta' \Rightarrow \alpha \alpha_i \beta', \ \alpha_i \beta' \overset{L}{\Rightarrow} a\gamma', \ \beta', \gamma' \in (V \cup \#)^*\}. \tag{2.1}$$

Azt mondjuk, hogy az A nemterminális szimbólum osztott, ha

$$[A, \alpha_i] \cap [A, \alpha_j] = \emptyset \quad \text{minden } i, j\text{-re}, \quad 1 \leqq i < j \leqq l. \tag{2.2}$$

A G KFN-t LF-nyelvtannak nevezzük, ha minden nemterminális szimbóluma osztott.

A LF-nyelvtan által generált nyelvet LF-nyelvnek nevezzük.

A fenti definíció megfelel a [4]-ben adottnak. Egy alternatíva baldeterminiális halmaza „kétfajta" szimbólumból épül fel. Egyrészt az alternatívából levezethető legbaloldalibb terminálisokból, másrészt ha az alternatívából levezethető az üres

string, akkor a valamely levezetésben az alternatívát követő terminálisból. Ennek pontos megfogalmazására vezessük be a következő jelöléseket:

Legyen G KFN és $\alpha \in V^*$.

$$E_0(\alpha) = \{a \in T \mid \alpha \overset{L}{\Rightarrow} *a\beta\}$$

$$E(\alpha) = \begin{cases} E_0(\alpha), & \text{ha} \quad \alpha \not\Rightarrow *\varepsilon \\ E_0(\alpha) \cup \{\varepsilon\} & \text{különben.} \end{cases} \tag{2.3}$$

$$K(A) = \{a \in T' \mid S \# \Rightarrow *\alpha A a \beta',\ \beta' \in (V \cup \#)^*\}$$

(2.1)-ből és (2.3)-ból következik, hogy

$$[A, \alpha_i] = \begin{cases} E(\alpha_i), & \text{ha} \quad \alpha_i \not\Rightarrow *\varepsilon, \\ (E(\alpha_i) \setminus \{\varepsilon\}) \cup K(A), & \text{különben.} \end{cases} \tag{2.4}$$

Egy nemterminális szimbólum LF-tulajdonságát általában nem közvetlenül a (2.2) formula szerint ellenőrizzük, hanem a fenti két halmaz különálló kiszámítása után a következő tétel a) és b) feltétele szerint:

2.1. Tétel. Egy $G=(N, T, S, P)$ redukált KFN akkor és csak akkor LF-nyelvtan, ha tetszőleges $A \to \alpha_1 | \alpha_2 | \ldots \alpha_l \in P$ esetén

a) $E(\alpha_i) \cap E(\alpha_j) = \emptyset$, $1 \leqq i < j \leqq l$-re;
b) Ha $\alpha_i \Rightarrow *\varepsilon$, akkor minden $j \neq i$-re, $1 \leqq j \leqq l$, $E(\alpha_j) \cap K(A) = \emptyset$.

[1]-ben A. V. Aho megfogalmazott egy hasonló tételt az LL(1)-nyelvtanokra.

Bizonyítás.

A bizonyításhoz rövidség kedvéért a következő jelöléseket vezetjük be:

$$E_i = E(\alpha_i),$$

$$E_j = E(\alpha_j),$$

$$C = \{\varepsilon\},$$

$$K = K(A).$$

Ennek megfelelően (2.4)-et úgy írhatjuk, hogy

$$[A, \alpha_i] = \begin{cases} E_i, & \text{ha} \quad \alpha_i \not\Rightarrow *\varepsilon, \\ (E_i \setminus C) \cup K, & \text{különben.} \end{cases}$$

Nyilvánvaló, hogy

$$[A, \alpha_i] \cap [A, \alpha_j] =$$

$$= \big((E_i \mid \alpha_i \not\Rightarrow *\varepsilon) \cup ((E_i \setminus C) \cup K \mid \alpha_i \Rightarrow *\varepsilon)\big) \cap \big((E_j \mid \alpha_j \not\Rightarrow *\varepsilon) \cup ((E_j \setminus C) \cup K \mid \alpha_j \Rightarrow *\varepsilon)\big),$$

azaz

$$(2.5)\quad [A,\alpha_i]\cap[A,\alpha_j]=\begin{cases} \text{i) } E_i\cap E_j, \text{ ha } \alpha_i \not\Rightarrow *\varepsilon \text{ és } \alpha_j \not\Rightarrow *\varepsilon, \\ \text{ii) } ((E_i\setminus C)\cup K)\cap E_j=((E_i\setminus C)\cap E_j)\cup(K\cap E_j), \\ \qquad\qquad \text{ha } \alpha_i \Rightarrow *\varepsilon \text{ és } \alpha_j \not\Rightarrow *\varepsilon, \\ \text{(Itt } (E_i\setminus C)\cap E_j=E_i\cap E_j, \text{ mivel } E_j \text{ nem tartalmazza } \varepsilon\text{-t}, \\ \quad \text{tehát } ((E_i\setminus C)\cup K)\cap E_j=(E_i\cap E_j)\cup(K\cap E_j).) \\ \text{iii) } ((E_j\setminus C)\cup K)\cap E_i=(E_i\cap E_j)\cup(K\cap E_i), \\ \qquad\qquad \text{ha } \alpha_i \not\Rightarrow *\varepsilon \text{ és } \alpha_j \Rightarrow *\varepsilon, \\ \text{iv) } ((E_i\setminus C)\cup K)\cap((E_j\setminus C)\cup K)=((E_i\cap E_j)\setminus C)\cup K, \\ \qquad\qquad \text{ha } \alpha_i \Rightarrow *\varepsilon \text{ és } \alpha_j \Rightarrow *\varepsilon. \end{cases}$$

A feltétel szükséges.

Ha G LF-nyelvtan, akkor az a) és b) feltétel igaz. Az LF-nyelvtan definíciója szerint $[A,\alpha_i]\cap[A,\alpha_j]=\emptyset$ teljesül bármely $A\in N$-re, bármely $i\neq j$-re. Ebből következik, hogy

i) $E_i\cap E_j=\emptyset$, ha $\alpha_i \not\Rightarrow *\varepsilon$ és $\alpha_j \not\Rightarrow *\varepsilon$,

ii) $E_i\cap E_j=\emptyset$,

$K\cap E_j=\emptyset$, ha $\alpha_i \Rightarrow *\varepsilon$ és $\alpha_j \not\Rightarrow *\varepsilon$,

iii) $E_i\cap E_j=\emptyset$,

$K\cap E_i=\emptyset$, ha $\alpha_i \not\Rightarrow *\varepsilon$ és $\alpha_j \Rightarrow *\varepsilon$.

iv) $(E_i\cap E_j)\setminus C=\emptyset$,

$K=\emptyset$, ha $\alpha_i \Rightarrow *\varepsilon$ és $\alpha_j \Rightarrow *\varepsilon$.

Mivel tetszőleges redukált nyelvtanban nyilván $K\neq\emptyset$, ezért a iv) eset nem állhat fenn. Az i)—iii) esetek első összetevőiből következik, hogy az a) feltétel igaz, míg az ii)—iii) esetek második feltételéből a b) feltétel igaz.

A feltétel elégséges

Ha G minden nemterminális szimbólumára teljesül az a) és b) feltétel, akkor G LF-nyelvtan. E pont bizonyításában is a (2.5) formulára fogunk támaszkodni. Az a) feltételből következik, hogy $\alpha_i \Rightarrow *\varepsilon$ és $\alpha_j \Rightarrow *\varepsilon$ egyszerre nem teljesülhet, tehát (2.5)-ben csak az első három esetet kell vizsgálni.

Az i) esetben $E_i\cap E_j\neq\emptyset$, az a) feltételből.

Az ii) esetben az egyesítés első tagja üres az a) feltétel miatt, a második pedig a b) feltétel szerint, így egyesítésük is üres. Ugyanez áll a iii) esetre.

Összegezve (2.5) szerint $[A,\alpha_i]\cap[A,\alpha_j]=\emptyset$, amit bizonyítani akartunk. A tételből könnyen belátható az LF-nyelvtan két ismert tulajdonsága [2, 4].

1. KÖVETKEZMÉNY. G LF-nyelvtan esetén egy nemterminális szimbólumnak legfeljebb egy olyan levezetése lehet, amelyik az üres stringet generálja, azaz ha

$A \to \alpha_1 | \alpha_2 \in P$, akkor $\alpha_1 \Rightarrow * \varepsilon$ és $\alpha_2 \Rightarrow * \varepsilon$ egyszerre nem állhat fenn. (Lásd a 2.1 tétel bizonyításának első részét.)

2. KÖVETKEZMÉNY. G LF-nyelvtan nem tartalmaz balrekurzív nemterminális szimbólumot.

Tekintsük az $A \to \alpha_1 | \alpha_2 \in P$ helyettesítési szabályt. Tegyük fel, hogy itt α_1 eredményez balrekurzív levezetést. Ekkor $A \Rightarrow \alpha_1 \Rightarrow * A\beta \Rightarrow \alpha_2\beta$, ahol $\beta \in V^*$. Amennyiben $\alpha_1 \not\Rightarrow * \varepsilon$, $\alpha_1 \Rightarrow * \alpha_2\beta$ miatt $E(\alpha_1) \supseteq E(\alpha_2)$, azaz $E(\alpha_1) \cap E(\alpha_2) \neq \emptyset$ és nem tesz eleget a 2.1. tétel a) feltételének.

Ha viszont $\alpha_2 \Rightarrow * \varepsilon$, akkor $\alpha_1 \Rightarrow * \beta$ és két lehetőség áll fenn:

— Ha $\beta \Rightarrow * \varepsilon$, akkor $\alpha_1 \Rightarrow * \varepsilon$. Ez ellentmond az 1. következménynek.

— Ha $\beta \not\Rightarrow * \varepsilon$, akkor $E(\alpha_1) \supseteq E(\beta)$ és $E(\beta) \subseteq K(A)$, azaz $E(\alpha_1) \cap K(A) \neq \emptyset$, és G nem tesz eleget a 2.1 tétel b) feltételének. Mindhárom esetben ellentmondásra jutottunk, így a 2. következményt bebizonyítottuk.

Mint említettük, e két tulajdonságot már korábban is bizonyították. A balrekurzió kérdését tárgyalja GRIFFITHS is, de pontatlanul jár el (Lásd [2] 6. oldal). Tekintsük például az $A \to Ab | \varepsilon$ nyelvtant. GRIFFITHS jelölésében

$$\alpha_1 = Ab, \qquad \alpha_2 = \varepsilon,$$

$$S(\alpha_1) = \{b\}, \quad S(\alpha_2) = \emptyset,$$

és így ebben az esetben $S(\alpha_1) \supseteq S(\alpha_2)$-ből (az eredetiben a tartalmazás jel nyilván nyomdahiba következtében fordítva szerepel) $S(\alpha_1) \cap S(\alpha_2) \neq \emptyset$ nem következik.

A fenti nyelvtan valóban nem LF (azaz LL (1)), de ez csak a következő szimbólumok figyelembevételével mutatható meg.

A bemutatott pontatlansággal kapcsolatban meg kívánjuk jegyezni, hogy az osztott nyelvtanok elméletének vizsgálatában az üres alternatíva gyakran okoz problémát. Így a WOOD által a balterminális halmaz felépítésére adott algoritmus hibája (lásd [5]) is az üres halmazzal függ össze.

Ismeretes, hogy a környezetfüggetlen nyelvtanok osztott formára átalakítása esetén az üres alternatíva a többitől eltérő kezelést igényel (lásd [4, 5, 6]). Ezek közül legáltalánosabb a [6], de nem fordít elég figyelmet az üres alternatívára. Ezt mutatja az $S \to abSa | \varepsilon$ nyelvtan, amely eleget tesz a [6]-ban kimondott 2. tétel feltételeinek, de rajta a tételben leírt átalakítás nem végezhető el.

Az [5]-ben leírt nyelvtanátalakító algoritmus egy nyelvtan osztott voltának ellenőrzését az

$$E(\alpha_i) \cap E(\alpha_j) = \emptyset, \quad i \neq j\text{-re} \tag{2.6}$$

$$\text{és} \quad E(A) \cap K(A) = \emptyset, \quad A \in N$$

összefüggések segítségével vizsgálja. A vizsgálat jogosságát azonban nem bizonyítja. Az alábbi tétellel megmutatjuk, hogy a (2.6) vizsgálat jogos.

2.2. TÉTEL. Egy $G = (N, T, S, P)$ redukált KFN akkor és csak akkor LF-nyelvtan, ha tetszőleges $A \to \alpha_1 | \alpha_2 | \ldots | \alpha_l \in P$ esetén

a) $E(\alpha_i) \cap E(\alpha_j) = \emptyset$, $1 \leqq i < j \leqq l$-re;

b) Ha $A \Rightarrow * \varepsilon$, akkor $E(A) \cap K(A) = \emptyset$.

Bizonyítás

A 2.1. tételt figyelembe véve nyilván csak azt kell bizonyítanunk, hogy ha egy LF-nyelvtan valamely $A \to \alpha$ szabályára igaz $\alpha \Rightarrow * \varepsilon$, akkor $E(\alpha) \cap K(A) = \emptyset$. Indirekt bizonyítást alkalmazunk. Tegyük fel, hogy

$$E(\alpha) \cap K(A) \neq \emptyset.$$

Mivel $\varepsilon \notin K(A)$ és $\# \notin E(\alpha)$ (lásd (2.3)-at), van olyan $a \in T$, hogy $a \in E(\alpha)$ és $a \in K(A)$. Ezek szerint $\alpha \overset{L}{\Rightarrow} * a\beta$ valamilyen $\beta \in V^*$-ra, valamint $\alpha \Rightarrow * \varepsilon$ miatt nyilván $\alpha \overset{L}{\Rightarrow} * \varepsilon$, tehát léteznek a

$$D_1 = \langle \alpha = \gamma_0, \gamma_1, \ldots, \gamma_{l_1} = \varepsilon \rangle,$$

$$D_2 = \langle \alpha = \delta_0, \delta_1, \ldots, \delta_{l_2} = a\beta \rangle \quad \text{bal-levezetések.}$$

Legyen $k = (\max i | \gamma_i = \delta_i)$. Jelöljük γ_k-t φ-vel; $\varphi \Rightarrow * \varepsilon$ alapján $\varphi \in N^*$, továbbá $\varphi \overset{L}{\Rightarrow} * a\beta$ miatt $\varphi \in N^+$, tehát $\varphi = B\varrho$, ahol $B \in N$ és $\varrho \in N^*$. Figyelembe véve, hogy D_1 és D_2 bal-levezetések, $\gamma_{k+1} = \beta_1 \varrho$, $\delta_{k+1} = \beta_2 \varrho$ és $\gamma_{k+1} \neq \delta_{k+1}$ miatt β_1 és β_2 a B nemterminális szimbólum két különböző alternatívája.

$\gamma_{l_1} = \varepsilon$-ból $\beta_1 \Rightarrow * \varepsilon$ és mivel G LF-nyelvtan, $\beta_2 \not\Rightarrow * \varepsilon$. Ezt figyelembe véve, $\beta_2 \varrho \Rightarrow * a\psi$-ből $a \in E(\beta_2)$ következik. Ezenkívül $\gamma_{l_1} = \varepsilon$-ból $\varrho \Rightarrow * \varepsilon$, úgy $A \Rightarrow \alpha \Rightarrow * B\varrho \Rightarrow * B$, és $K(A) \subseteq K(B)$, azaz $a \in K(B)$.

Összegezve $B \to \beta_1 | \beta_2 \in P$, ahol $\beta_1 \neq \beta_2$; $\beta_1 \Rightarrow * \varepsilon$ és $E(\beta_2) \cap K(B) \neq \emptyset$, ami ellentmond a 2.1. tételben bizonyított b) feltételnek. Így a 2.2. tételt bebizonyítottuk. A 2.2. tétel abban tér el lényegesen az előbbitől, hogy míg a 2.1. tétel a nyelvtan egy nemterminális szimbólumára különállóan is igaz, addig az utóbbi nem. Például a következő nyelvtanban:

$$S \to aAb,$$

$$A \to B | cBb,$$

$$B \to b | \varepsilon.$$

Nyilvánvalóan az A nemterminális szimbólum osztott, de

$$E(A) \cap K(A) = \{b\} \neq \emptyset.$$

Végezetül köszönetünket fejezzük ki Dömölki Bálintnak és Komor Tamásnak, akik tanácsaikkal segítették munkánkat.

IRODALOM

[1] Aho, A. V. and Ullman, J. D., *The Theory of Parsing, Translation and Compiling, Volume 1: Parsing* (Prentice-Hall Inc., 1972).

[2] Griffiths, M., *LL(1) Grammars and Analysers. (Advanced course on compiler construction)* Lecture Note, Techn. Univ. Münich, Germany, 1974. (Magyarul: Számológép Kiskönyvtár 1975).

[3] Komor, T. és Vu-Luc, Határozott levezető elemzés, Programozási Rendszerek'75 Konferencia Kötet, Szeged, 1975.

[4] Wood, D., "The theory of left factored languages, Part 1 and Part 2", *The Computer Journal*, **12** (1969) 349—356 and **13** (1970) 55—62.

[5] Комор, Т., «О преобразовании грамматик к разделённой форме», Kandidátusi disszertáció 1973.
[6] Синдеев, В. Р., «Алгоритм преобразования синтаксиса к виду удобному для однопроходной трансляции», Програмирование 4 (1975).

(Beérkezett: 1976. április 29.)

(Újra beérkezett: 1976. november 25.)

VU-LUC
SZÁMÍTÓGÉPALKALMAZÁSI KUTATÓ INTÉZET
1015 BUDAPEST I., CSALOGÁNY U. 30—32.

ON NECESSARY AND SUFFICIENT CONDITIONS OF THE LF GRAMMARS

VU-LUC

In practical applications of the LF method for syntactical analysis an alternative definition of the LF grammars, which seems to be more convenient in terms of computability than the one given originally by Wood, is frequently used. The equivalence of these definitions is proved in this paper.

Alkalmazott Matematikai Lapok 2 (1976) 315–329

SZÁMÍTÓGÉPHÁLÓZATOK ADATKAPCSOLAT SZINTŰ PROTOKOLLJÁNAK FORMÁLIS DEFINÍCIÓJA

HARANGOZÓ JÓZSEF
Budapest

A számítógéphálózatok kommunikációs protokolljai egy hierarchikus többszintű rendszert alkotnak. A dolgozat e hierarchikus rendszer egyetlen szintjének, az ún. adatkapcsolat szintnek a logikai analízisével foglalkozik. Kísérletet tesz egy formális nyelv segítségével a *High Level Data Link Control Procedure* (HDLC) egy részhalmazának formális leírására. A távoli gépek közötti „párbeszéd" adatkapcsolat szinten ún. információs, felügyelő és számozatlan *frame*-ek cseréjén keresztül valósul meg. E *frame*-ek egy szimbólumsorozat elemeinek tekinthetők. A *frame*-ek sorozata, tehát a „párbeszéd", mondat-struktúrát alkot, amely — BJORNER és HOFFMANN szerint — reguláris nyelvtannal leírható. A dolgozat bemutat egy olyan módszert, amely segítségével a számítógépek közötti „párbeszéd" adatkapcsolat szinten reguláris nyelvtannal generálható. Többszintű felosztással további reguláris nyelvtanok szerkeszthetők az alacsonyabb szintű folyamatok leírására. Bemutatásra kerül egy ARPA-típusú *frame*-szerkezet és a hozzá tartozó generáló nyelvtan is, amely a KFKI intézeti hálózatában kerül implementálásra.

1. Bevezetés

A számítógépen belüli folyamatok koordinálása az operációs rendszer feladata, amelyet többnyire az ún. *process management system*, azon belül pedig a *process scheduler* végez [1]. Hasonló probléma merül fel számítógépeknek egymáshoz történő kapcsolásakor, amikoris két, egymástól független folyamat együttfutását, koordinálását kell elvégezni. Ahhoz, hogy az egymástól távol elhelyezkedő két gép folyamatai között kapcsolatot lehessen teremteni, egy sor megegyezésre, szabályra van szükség. E szabályok halmaza — a *protokoll* — fogja meghatározni a két folyamat interakciójának menetét, vagyis azt, hogy melyik folyamat, mikor, milyen formában küldhet üzenetet a másik folyamatnak vagy fogadhat üzenetet a másik folyamattól. Minthogy a folyamatok közötti kapcsolatteremtéshez az adatátviteli hálózatot is fel kell használni, a fizikai kapcsolatfelvétel elengedhetetlen, így a távoli folyamatok interakciója több szinten keresztül valósul meg (pl.: user, logikai, adatkapcsolat[1], fizikai szinten). Természetesen a szabályok egy-egy halmaza — a protokollok rendszere — is tükrözi ezt a többszintű hierarchikus felépítést.

A dolgozat ezen hierarchikus protokollrendszer egyetlen szintjét, az ún. adatkapcsolat szintet veszi vizsgálat alá. Az adatkapcsolat szintű protokollok tartalmazzák a hálózat fizikai működésének vezérlő eljárásait és a vezérlés formáit. Az adatkapcsolat szintű protokollra épül a fölötte levő logikai szint. Alapkövetelmény

[1] *Adatkapcsolat* (*data link*): két vagy több adatállomás közötti információcserét lehetővé tevő fizikai összeköttetés.

a szomszédos szintek kapcsolatában, hogy a felül levő szint számára az alatta levő szint transzparens legyen.

A dolgozat második fejezete röviden ismerteti a vezérlésre szolgáló eljárást. A következő fejezetben rámutatunk a protokollok formális leírásának szükségességére. A negyedik fejezet ezzel kapcsolatban néhány eddig alkalmazott módszert mutat be. A kidolgozott új módszert az ötödik fejezet tartalmazza. A következő fejezet a KFKI-ban implementálásra kerülő módosított HDLC-eljárás formális leírását mutatja be. Az utolsó fejezet a bemutatott módszer rövid értékelését tartalmazza.

2. A vizsgált protokoll

A dolgozat az *International Standardization Organization* (ISO) által kidolgozott és ajánlott adatkapcsolat szintű protokollt, a *High Level Data Link Control Procedure* (HDLC), egyik részhalmazát veszi vizsgálat alá, s az ún. „*normal response mode*" (NRM)[2] és félduplex átvitel (HDX)[3] esetén követendő eljárások [2] formális leírására kíván egy módszert bemutatni.

A jelzett ajánlás szerint, az egymással kapcsolatban álló két állomás közül az egyiket *primary*, a másikat *secondary* állomásnak nevezzük. A *primary* parancsokat, a *secondary* válaszokat küld. A parancsok és a válaszok egy-egy nemüres, véges halmaz elemei. Mind a parancsok, mind a válaszok formátuma kötött, és az ún. *frame-struktura* alapján képződnek [3]. Az ISO-ajánlás háróféle *frame-típust* definiál: információs vagy I-*frame*-et, felügyelő vagy S-*frame*-et és számozatlan vagy U-*frame*-et.

Az I-*frame* felépítése a következő:

flag mező (8)	cím mező (8)	vezérlő mező (8)	információs mező (n)	ellenőrző mező (16)	*flag* mező (8)

A *flag* mező egy előre meghatározott *bit-kombináció* (01111110), amely a *frame* kezdetét és végét jelöli. A cím a *secondary* címe. A vezérlő mező egy *frame-típus kódot,* a legközelebb elküldendő és a várt *frame* sorszámát, és egy *poll/final bitet* tartalmaz. Az információs mezőben tetszőleges számú adatbit helyezkedhet el. Az ellenőrző mező tartalmazza a *ciklikus redundancia* (CRC)-*képző áramkör* által a *flag* utáni mezőkre végzett számlálásának eredményét. A zárójelbe tett számok a mezőben levő bitek számát mutatják. Egy *I-frame* parancs is, válasz is lehet.

Az *S-frame* szerkezete a következő:

flag mező (8)	cím mező (8)	vezérlő mező (8)	ellenőrző mező (16)	*flag* mező (8)

[2] *Normal response mode:* olyan működési mód, amelyben a *secondary* állomás csak a *primary* állomástól jövő felszólítás után válaszolhat.

[3] *Félduplex átvitel:* az átviteli vonalon egy adott időpillanatban csak egyirányú információáramlás van.

Ez a *frame* nem tartalmaz információt, csupán vezérlést, felügyeletet lát el. A parancs vagy válasz típusa a vezérlő mezőből olvasható ki, amely a *frame-típus kódot*, a parancs kódot, a várt üzenet sorszámát és egy *poll/final bitet* tartalmaz. Az *S-frame* lehet parancs is és válasz is: *Receive Ready* (RR) és *Receive Not Ready* (RNR).

Az *U-frame* felépítése megegyezik az *S-frame* felépítésével, csupán a vezérlő mező szerkezete különbözik, s így a képzett parancsok és válaszok is mások. Parancsok a következők: *Set Normal Response Mode* (SNRM) és DISC*onnect* (DISC), míg a válaszok: *Unnumbered Acknowledgement* (UA) és *ComManD Reject* (CMDR).

A két állomás közötti kapcsolat három fázisra bontható: (1) a kapcsolat felépítése, (2) az adatátvitel, (3) a kapcsolat lebontása. A kapcsolat felépítést a *primary* állomás kezdeményezi egy SNRM paranccsal, amelyre a *secondary* vagy UA vagy CMDR választ küld attól függően, hogy megértette-e a parancsot és hajlandó a kapcsolatfelvételre, vagy visszautasítja azt. Pozitív válasz esetén a *primary* vagy elkezdi az adatátvitelt *I-* vagy egy *S-frame* küldésével (RR) a *secondary*t szólítja fel adásra. Ha a *secondary*nak van küldendő adata, akkor elküldi (I), ha nincs akkor egy *S-frame* küldésével (RR) a vezérlés a *primary*hez kerül, és a folyamat I vagy RR küldésével folytatódik. Egy *I-frame* küldése utáni foglaltságot RNR parancs vagy válasz küldésével jelzik egymásnak. Ha a *primary* a kapcsolat bontását kívánja, akkor egy DISC parancsot továbbít, amelyre a *secondary* vagy UA vagy CMDR választ ad. UA esetén a két berendezés közötti kapcsolat megszakad.

3. A formális leírás szükségessége

A számítógéphálózatok protokolljait — így a fenti protokollt is — sokkal jobban kézben lehetne tartani, könnyebb lenne mind a tervezése, mind pedig az implementálása egy adott rendszerben, ha valamilyen módon formálisan is le lehetne írni. A protokollokat mindezideig valamilyen élő nyelven — legtöbb esetben angol nyelven — tették közzé. Ezek a protokoll-leírások igen bonyolult nyelvezetűek, hogy lehetőleg minél pontosabban kövessék a lejátszódó folyamatokat és meghatározzák a szükséges feltételeket. Ennek ellenére implementáláskor mégis számos kétértelműség, egyéni értelmezés akadályozza az egységes megvalósítást, amely különösen akkor jelent komoly gondot, ha e munkát nem egy, hanem több csoport végzi a különböző gépeken.

Szükséges a formális leírás akkor is, ha egymáshoz többé-kevésbé hasonló protokollokat szeretnénk összehasonlítani. Sokkal jobban látszanak az egyes változatok előnyös és hátrányos tulajdonságai, s így a rendszert tervező egyértelműen eldöntheti, melyik a számára legmegfelelőbb eljárás.

Formális leírás fényt vet a szöveges leírásból esetleg közvetlenül ki nem derülő patthelyzetekre is *(dead-lock)*. Sokkal jobban kézben tarthatók az egymással kapcsolatban álló számítógépekben lejátszódó folyamatok.

A formális leírásra szükség van a protokollnak, mint digitális rendszernek *a tervezéséhez* is.

Digitális rendszerek logikai tervezéséhez ugyanis két, egymással szorosan kapcsolódó követelményt kell kielégíteni:

— a digitális rendszer formális definiálása,
— a formális leírás mechanikus leképezése a rendszer logikai függvényébe.

Az így kapott logikai függvény azután akár *hardware*, akár *software* vagy *firmware* úton implementálható [6].

A protokollok *implementálását* szintén megkönnyíti a formális leírás. Segítségével megvilágíthatók az egyes protokoll-szinteken lejátszódó események, valamint az egyes szintek közötti viszony. Ha egy ilyen formális leírás rendelkezésünkre áll, az implementálás egyértelmű és világos megoldást eredményez.

4. Néhány módszer a protokollok formális leírására

A protokollok formális leírásának gondolata már régóta foglalkoztatja a számítógéphálózat tervezésével, implementálásával foglalkozó szakembereket. Több módszert dolgoztak ki, melyek közül néhányat a teljesség igénye nélkül bemutatunk.

CERF, az ARPA-hálózat egyik ismert tervezője egy gráfelméleti módszert dolgozott ki a számítógépen belüli számítás modellezésére [4]. Az ún. UCLA *Graph Model* közvetlenül még nem volt alkalmas a protokollok tanulmányozására, de jó alapnak bizonyult a további kutatások számára.

POSTEL a CERF által kidolgozott UCLA *Graph Model*-t továbbfejlesztette [5]. Kimutatta, hogy a *Cerf-féle modell* és a *Petri-háló* egymásnak megfeleltethető, majd a *Cerf-modell* protokollok leírására történő átalakításával és gráf-modulok képzésével egy viszonylag egyszerű modell megalkotását javasolta. Annak ellenére, hogy e módosított változat a bemutatott példákon jól funkcionál, kissé bonyolultnak tűnik, különösen ha a valóságos viszonyokat tekintjük.

BJORNER [6] vizsgálati módszere az ún. többszintű iteráción alapul, melynek során a vizsgált protokollt először mint egy teljes rendszert írja le, majd megvizsgálva a rendszer komponenseit alrendszereket épít fel, míg végül megkapja a teljes protokollszint formális leírását egy működési gráfba sűrítve. E módszert más területeken is alkalmazzák. Előnye, hogy a teljes protokoll-szint működése a kapott gráfról jól követhető.

RUSBRIDGE és LANGSFORD [7] abból indult ki, hogy a protokoll nem más, mint azoknak a szabályoknak a halmaza, amelyet az egymással kapcsolatban álló két számítógépben lezajló folyamatoknak figyelembe kell venni. Ha az egyes folyamatok formális képe egy-egy automata, a protokoll tulajdonképpen e két automata egymással való párbeszédét írja le, s így a két automata soros kompozíciójával formálisan is megadható. Egymástól távol elhelyezkedő gépek esetén az adatátviteli vonal, mint egy meglehetősen passzív automata hatását is figyelembe kell venni. A módszerrel készült modell nagyon jól követi a valóságos viszonyokat csupán a kompozit automata lesz kissé bonyolult konstrukció.

MERLIN hasonló gondolatmenetet követ, mint a fenti szerzők, ő is egymástól távoli két folyamat egymásközötti kapcsolatát vizsgálja [8]. A választott eszköz azonban nem automata, hanem a *Petri-háló*. Módszerével a protokollok ún. „*recoverability*" tulajdonságát vizsgálja, vagyis hogy a hibás működésből hogyan tud a teljesítőképesség csökkenése nélkül kikerülni. A bemutatott módszer az ún. T*ime*-P*etri*-N*et* (TPN) modell képes figyelembe venni a protokoll által meghatározott időzítéseket is. E módszer további tökéletesítéssel minden bizonnyal alkalmas lehet az adatátviteli késleltetések figyelembevételére is.

HOFFMANN a formális nyelvek oldaláról elindulva próbál módszert adni a protokollok formális leírására [9]. Megfigyelve az átviteli vonalakat egy átviteli eljárás

során, azt tapasztaljuk, hogy karaktersorozatok áramlanak egymás után. E sorozatok szabályossággal rendelkeznek, s valamilyen formális nyelvtannal generálhatók. Megadható egy olyan nyelvtan, amely az adási és vételi folyamatot követi, figyelembe veszi az időzítés és a ciklikus redundancia képzés hatását is. A módszer jól közelíti meg a problémát, de az adási és vételi folyamat részletein túl az interakció magasabb szintjeit nem képes követni.

Az itt felsorolt módszerek egy része hasznos segítő eszköz lehet a protokollok formális leírásában, más része viszont korlátozásokat vesz figyelembe a protokoll leírásakor, mivel egy korábbi, egészen más felépítésű *link* szintű protokoll leírására dolgozták ki, s magán viseli annak a protokollnak jegyeit. A HDLC-eljárás formális leírását a fenti módszerek segítségével a szerző is megkísérelte. A következőkben egy olyan módszert kívánunk bemutatni, amely általános, konkrét protokolltól független formális leírását adhatja egy adott szinten végbemenő eseményeknek.

5. A kidolgozott módszer

A bemutatásra kerülő módszert néhány fenti módszer alapkoncepciójának figyelembevételével dolgoztuk ki. A protokoll nem más, mint két berendezés közötti dialógus szabályait leíró halmaz. Ha sikerül megkonstruálni azokat a halmazokat, amelyek elemeiből képezik az egyes berendezések szimbólum-sorozataikat, valamint sikerül megkonstruálni azt a nyelvtant, amely jól leírja azokat a szabályokat, amelyek alapján az egyes berendezések e sorozatokat generálják, tulajdonképpen a protokoll formális leírását már megadtuk.

A tervezés és implementálás érdekében az így kapott nyelvtant részekre kell bontani, s további nyelvtanokat kell konstruálni a részfolyamatok leírására.

A konstrukciós algoritmus a következő:

(1) A „párbeszéd"-et leíró nyelvtan megkonstruálása:
   a) Határozzuk meg azon szimbólumok halmazát, amelyet az egyik, és azokét, amelyet a másik berendezés küld a vonalra. E két halmaz uniója képezi a vonalon áramló szimbólumok, vagyis a terminálisok halmazát.
   b) A nemterminálisok halmazának megadása.
   c) Képezzük ezután megfelelő módon — a szöveges leírás, vagy más elképzelés gondolatmenete alapján — a produkciós szabályok halmazát.
(2) A „párbeszéd" folyamat felbontása alacsonyabb működési szintekre.
(3) Az így kapott alacsonyabb szintekhez generáló nyelvtan konstruálása az (1) pontban ismertetett módon.
(4) A szomszédos szintek közötti *interface* definiálása.
(5) Egy rendszer-nyelvtan konstruálása, amely figyelembe veszi mind a különböző szintek jellemzőit, mind a köztük levő kapcsolatot.

A továbbiakban nézzük meg néhány példán, hogy e módszert hogyan lehet a korábban ismertetett protokoll formális leírására felhasználni.

Generáló nyelvtan a „párbeszéd" leírására

A HDLC-eljárás „*normal response mode*" és félduplex átvitel esetén követett működését már a második fejezetben ismertettük, így most megkíséreljük ennek a formális leírását adni a fenti módszer segítségével.

Legyen $G^1=(V_N^1, V_T^1, P^1, S^1)$ egy nyelvtan, amelynek elemei a következők lesznek:

$V_T^1=V_{TP}^1 \cup V_{TS}^1$, ahol

$$V_{TP}^1 = \{\overrightarrow{SNRM}, \overrightarrow{DISC}, \overrightarrow{I}, \overrightarrow{RR}, \overrightarrow{RNR}\}$$

azon terminálisok halmaza, amelyet a *primary* küld a *secondary*nak,

$$V_{TS}^1 = \{\overleftarrow{UA}, \overleftarrow{CMDR}, \overleftarrow{I}, \overleftarrow{RR}, \overleftarrow{RNR}\}$$

azon terminálisok halmaza, amelyet a *secondary* küld a *primary*nek,

$\rightarrow$: jelzi az átviteli irányt a *primary*től a *secondary* felé,

$\leftarrow$: jelzi az átviteli irányt a *secondary*től a *primary* felé,

$$V_N^1 = \{S^1, A^1, B^1, C^1, D^1, E^1, F^1, H^1, J^1, K^1, L^1, M^1, N^1, O^1\}$$

P^1: a produkciós szabályokat az 1. táblázat tartalmazza (a $*$ jel a $H^1 \rightarrow \overline{UA}$ helyettesítési szabályt jelöli; a — jel jelentése: nincs reláció).

1. TÁBLÁZAT

Nemterminálisok	Terminálisok									
	$\overrightarrow{SNRM}$	$\overrightarrow{DISC}$	$\overrightarrow{I}$	$\overrightarrow{RR}$	$\overrightarrow{RNR}$	$\overleftarrow{UA}$	$\overleftarrow{CMDR}$	$\overleftarrow{I}$	$\overleftarrow{RR}$	$\overleftarrow{RNR}$
S^1	A^1	—	—	—	—	—	—	—	—	—
A^1	—	—	—	—	—	B^1	C^1	M^1	M^1	M^1
B^1	A^1	H^1	D^1	E^1	F^1	—	—	—	—	—
C^1	A^1	H^1	—	—	—	—	—	—	—	—
D^1	—	—	D^1	E^1	F^1	M^1	C^1	J^1	K^1	L^1
E^1	—	—	D^1	E^1	—	M^1	C^1	J^1	K^1	L^1
F^1	—	—	D^1	—	F^1	M^1	C^1	N^1	K^1	L^1
H^1	—	—	—	—	—	*	C^1	M^1	M^1	M^1
J^1	A^1	H^1	D^1	E^1	F^1	—	—	J^1	K^1	L^1
K^1	A^1	H^1	D^1	E^1	F^1	—	—	J^1	K^1	—
L^1	A^1	H^1	O^1	E^1	F^1	—	—	J^1	—	L^1
M^1	A^1	H^1	—	—	—	—	—	—	—	—
N^1	A^1	H^1	D^1	E^1	F^1	—	—	N^1	N^1	N^1
O^1	—	—	O^1	O^1	O^1	M^1	C^1	J^1	K^1	L^1

A produkciós szabályokból látható, hogy az így konstruált nyelvtan egy egyszerű reguláris nyelvtan, így a vele generált nyelv — vagyis a „párbeszéd" — egy 3. típusú nyelv.

A „párbeszéd" folyamatának dekompozíciója

A HDLC-protokoll „párbeszéd" folyamatát az alábbi alacsonyabb szintű folyamatokra lehet bontani:

(1) *Frame-generálási folyamat,* amely a *frame-szerkezet* különböző mezőkből (*fieldek*ből) történő összeállítását végzi. Az eljárás szabályainak megfelelően *flag,* cím, vezérlő, információs és ellenőrző mezőkből állítható össze egy *frame.*

(2) *Mező-generálási folyamat,* amely a mezőknek különböző részmezőkből történő összeállítását végzi.

(3) *Részmező-generálási folyamat,* amely a különböző rész-mezők előállítását végzi.

Ezen folyamat eredménye az a *bit*-sorozat, amely az adatátviteli vonalakra fog kerülni.

A fenti felbontásnak megfelelően konstruáljuk meg a generáló nyelvtanokat.

Generáló nyelvtan a frame előállításának leírására

A *frame* szerkezetét a második fejezetben, a protokoll leírásakor ismertettük. Feladatunk most az, hogy olyan nyelvtant konstruáljunk, amely mind a három típusú (*I, S, U*) *frame*-et előállítja.

Legyen $G^2=(V_N^2, V_T^2, P^2, S^2)$ a generáló nyelvtan, ahol

$$V_T^2=\{\textit{flag}, \text{cím, vezérlő, információ, ellenőrző}\}$$

$$V_N^2 = \{S^2, A^2, B^2, C^2, D^2\}.$$

P^2: a produkciós szabályokat a 2. táblázat tartalmazza. (A $*$ jel $X^2 \to a$ típusú helyettesítést jelöl, ahol $X^2 \in V_N^2$; $a \in V_T^2$.)

2. TÁBLÁZAT

Nemterminálisok	Terminálisok				
	flag	cím	vezérlő	információ	ellenőrző
S^2	A^2	—	—	—	—
A^2	—	B^2	—	—	—
B^2	—	—	C^2	—	—
C^2	—	—	—	D^2	$*$
D^2	—	—	—	—	$*$

A produkciós szabályok alapján megállapítható, hogy a nyelvtan reguláris, az általa generált nyelv — a *frame-előállítás folyamata* — 3. típusú.

A mezőgenerálás leírása

A felbontás következő szintjén, a mezőgeneráláskor a különböző típusú mezők előállítása folyik részmezőkből. A generálási folyamat egy $\mathcal{G}^3$ nyelvtanhalmazzal írható le:

$$\mathcal{G}^3 = \{G_S^3, G_C^3\}, \quad \text{ahol}$$

G_S^3 a *flag*, a cím, az információs és az ellenőrző mező generálására szolgáló, ezen a szinten rendkívül egyszerű nyelvtan, amely az alábbi szerkezetű:

$$G_S^3 = (\{S_S^3, A_S^3\}, \{\varepsilon\}, \{S_S^3 \to A_S^3\}, S_S^3);$$

G_C^3 a vezérlő mezőnek részmezőkből történő előállítását reprezentáló nyelvtan. A 3. táblázat a vezérlő mező felépítését mutatja, segítvén a 4. táblázatban megadott produkciós szabályok megértését.

3. TÁBLÁZAT

	Részmezők			
I-frame	*IC*	*NS*	*PF*	*NR*
S-frame	*SC*	*SV*	*PF*	*NR*
U-frame	*UC*	*M1*	*PF*	*M2*

A G_C^3 nyelvtan szerkezete a következő:

ahol

$$G_C^3 = (V_{NC}^3, V_{TC}^3, P_C^3, S_C^3),$$

$$V_{NC}^3 = \{S_C^3, A_C^3, B_C^3, C_C^3, D_C^3, E_C^3, F_C^3, H_C^3, J_C^3\}$$

$$V_{TC}^3 = \{IC, NS, PF, NR, SC, SV, UC, M1, M2\}$$

4. TÁBLÁZAT

Nemterminálisok	Terminálisok								
	IC	*NS*	*PF*	*NR*	*SC*	*SV*	*UC*	*M1*	*M2*
S_C^3	A_C^3	—	—	—	D_C^3	—	F_C^3	—	—
A_C^3	—	B_C^3	—	—	—	—	—	—	—
B_C^3	—	—	C_C^3	—	—	—	—	—	—
C_C^3	—	—	—	*	—	—	—	—	—
D_C^3	—	—	—	—	—	E_C^3	—	—	—
E_C^3	—	—	C_C^3	—	—	—	—	—	—
F_C^3	—	—	—	—	—	—	—	H_C^3	—
H_C^3	—	—	J_C^3	—	—	—	—	—	—
J_C^3	—	—	—	—	—	—	—	—	*

P^3: a produkciós szabályok a 4. táblázatban láthatók (a * jel a $X^3 \to a$ típusú helyettesítéseket jelöli, ahol $X^3 \in V_{NC}^3$; $a \in V_{TC}^3$).

Részmező- (bit-) generálás leírása

A részmezők generálásának szintjén a részmezőknek bitekkel történő előállítása folyik. A generálás eredménye az a bitsorozat, amely az átviteli vonalakon továbbításra kerül. A folyamatot egy $\mathscr{G}^4$ nyelvtanhalmazzal tudjuk leírni.

$$\mathscr{G}^4 = \{G_C^4, G_F^4, G_{IC}^4, G_{SC}^4, G_{UC}^4, G_{SV}^4, G_{M1}^4, G_{M2}^4, G_{PF}^4\}, \quad \text{ahol}$$

G_C^4 a cím, információs és ellenőrző mezők, valamint a vezérlőmező $N(S)$ és $N(R)$ részmezejének végleges formájú előállítását írja le. Szerkezete a következő:

$$G_C^4 = (\{S_C^4, A_C^4, B_C^4\}, \{0, 1\}, \{S_C^4 \to 0\, A_C^4;\ S_C^4 \to 1\, B_C^4;$$
$$A_C^4 \to 0\, A_C^4;\ A_C^4 \to 1\, B_C^4;\ B_C^4 \to 0\, A_C^4;\ B_C^4 \to 1\, B_C^4\},\ S_C^4);$$

G_F^4 a flag mező végleges formájú előállításának folyamatát tükrözi:

$$G_F^4 = (\{S_F^4, A_F^4, B_F^4\}, \{0, 1\}, \{S_F^4 \to 0\, A_F^4;\ A_F^4 \to 1\, B_F^4;\ B_F^4 \to 1\, B_F^4;\ B_F^4 \to 0\},\ S_F^4);$$

G_{IC}^4 a vezérlő mező *IC* részmezejének generálását írja le:

$$G_{IC}^4 = (\{S_{IC}^4\}, \{0\}, \{S_{IC}^4 \to 0\},\ S_{IC}^4);$$

G_{SC}^4 a vezérlő mező *SC* részmezejének generálását mutatja:

$$G_{SC}^4 = (\{S_{SC}^4, A_{SC}^4\}, \{0, 1\}, \{S_{SC}^4 \to 1\, A_{SC}^4;\ A_{SC}^4 \to 0\},\ S_{SC}^4);$$

G_{UC}^4 a vezérlő mező *UC* részmezejének generálását tükrözi:

$$G_{UC}^4 = (\{S_{UC}^4, A_{UC}^4\}, \{0, 1\}, \{S_{UC}^4 \to 1\, A_{UC}^4,\ A_{UC}^4 \to 1\},\ S_{UC}^4);$$

G_{SV}^4 a vezérlő mező *SV* részmezejének generálását írja le:

$$G_{SV}^4 = (\{S_{SV}^4, A_{SV}^4, B_{SV}^4\}, \{0, 1\}, \{S_{SV}^4 \to 0\, A_{SV}^4;\ S_{SV}^4 \to 1\, B_{SV}^4;\ A_{SV}^4 \to 0;\ B_{SV}^4 \to 0\},\ S_{SV}^4);$$

G_{M1}^4 a vezérlő mező *M*1 részmezejének generálását írja le:

$$G_{M1}^4 = (\{S_{M1}^4, A_{M1}^4\}, \{0\}, \{S_{M1}^4 \to 0\, A_{M1}^4,\ A_{M1}^4 \to 0\},\ S_{M1}^4);$$

G_{M2}^4 a vezérlő mező *M*2 részmezejének generálását tükrözi:

$$G_{M2}^4 = (\{S_{M2}^4, A_{M2}^4, B_{M2}^4\}, \{0, 1\}, \{S_{M2}^4 \to 0\, A_{M2}^4;\ S_{M2}^4 \to 1\, B_{M2}^4;\ A_{M2}^4 \to 1;\ B_{M2}^4 \to 0\},\ S_{M2}^4);$$

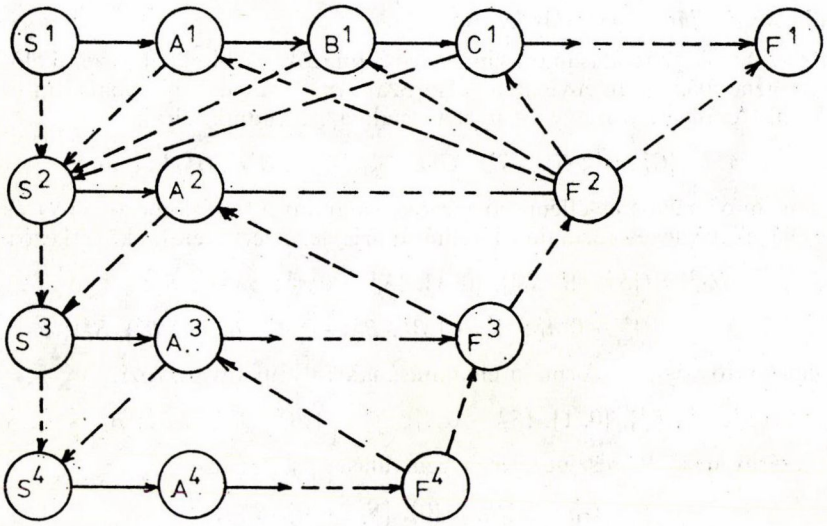
G_{PF}^4 a vezérlő mező *PF* részmezejének generálását mutatja:

$$G_{PF}^4 = (\{S_{PF}^4\}, \{0, 1\}, \{S_{PF}^4 \to 0;\ S_{PF}^4 \to 1\},\ S_{PF}^4).$$

Az interface definiálása

A különböző szintű folyamatokat leíró nyelvtanok meghatározása után a szomszédos szintek közötti *interface*-t is definiálni kell. Az *interface* megadásának elve a következő.

Két szomszédos szintet vizsgálva megállapíthatjuk, hogy miközben a magasabbik szinten egyetlen helyettesítés történik a nyelvtanban, az alacsonyabb szinten egy helyettesítési sorozat zajlik le, vagy fordítva, miközben az alacsonyabb szinten egy teljes helyettesítési sorozat lezajlik, s az ekvivalens véges automata végállapotba jut, addig a fölötte levő szinten csupán egyetlen helyettesítés megy végbe. Ez azt



1. ábra

jelenti, hogy ha definiálni tudunk transzformációs szabályokat a két szint között mindkét irányban, akkor ezek adják az interface formális képét. Megfigyeléseink alapján ezek a szabályok megkonstruálhatók az alábbi módon (1. ábra):

(1) minden helyettesítési szabály baloldali nemterminálisáról leképezés szükséges az alatta levő szint egyik nyelvtanának mondat-szimbólumára;

(2) minden szint nyelvtanának nemterminális halmazához egy pótlólagos nemterminálist kell hozzávenni, amely az ekvivalens véges automata végállapotának felel meg, s e nemterminálisról leképezést kell végrehajtani a fölötte levő szint nyelvtana helyettesítési szabályainak jobboldali nemterminálisaira.

Ezek a transzformációs szabályok reprezentálják a különböző szintű folyamatok közötti koordinációt.

A rendszer-nyelvtan előállítása

Az előzőekben megkonstruált, különböző szintű folyamatokat leíró nyelvtanok és a köztük levő interface megadása után előállítható az adott protokoll tervezése és implementálása szempontjából lényeges, ún. rendszer-nyelvtan.

$$G = (V_N, V_T, P, S)$$

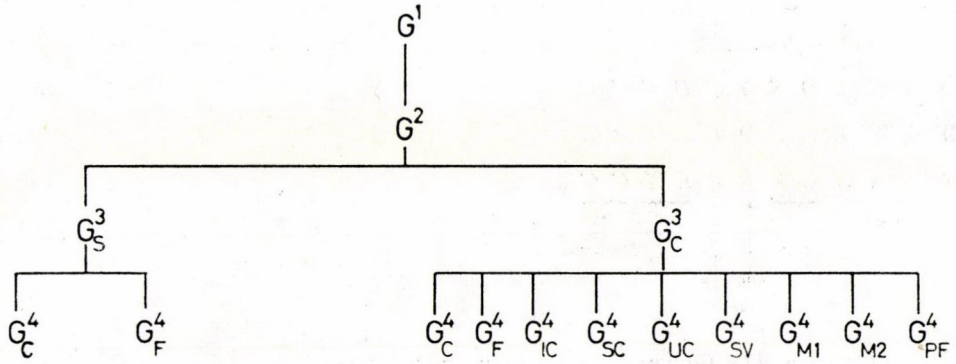
$V_N = \bigcup_i V_N^i \bigcup_i V_{NF}^i$, ahol i jelenti a különböző szintek sorszámát, és $i = 1, 2, 3, \ldots$; V_{NF}^i jelenti a nemterminálisok halmazához pótlólagosan hozzávett nemterminálisok halmazát.

$$V_T = \bigcup_i V_T^i$$

$$P = \bigcup_i P^i \bigcup_i \{(V_N^i \setminus F^i) \to S^{(i-1)}\} \bigcup_i \{(F^i \to (V_N^{(i+1)} \setminus S^{(i+1)}))\},$$

ahol $F^i \in V_{NF}^i$; $S^i \in V_N^i$.

A 2. ábrán a G rendszer-nyelvtan felépítése látható.



2. ábra

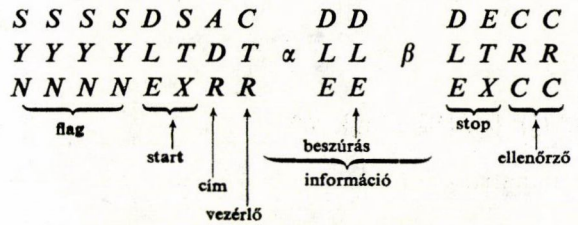
6. Generáló nyelvtan a KFKI-frame előállításának leírására

A KFKI-frame szerkezete

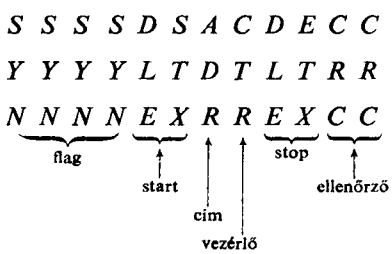
A KFKI-ban készülő számítógéphálózat protokolljának tervezésénél lehetőleg követni kívántuk a HDLC-előírásait, kihasználva azokat az előnyöket, melyeket ez az eljárás nyújt. *Hardware* megkötöttségek miatt azonban néhány helyen az eredeti ISO-ajánlást megvalósítani nem tudtuk, s annak egy módosított változatát használtuk fel. A módosítást a *frame* szerkezetében kellett elvégezni. A megváltoztatott *frame-szerkezet* nagyon hasonlít az ARPA-hálózatnál [10], valamint az IIASA-hálózatnál [11] alkalmazott konstrukcióra. Az eredeti HDLC *bit-orientált frame-szerkezete* helyett itt *karakter-orientált szerkezetet* használunk. A *frame* kezdetét és végét jelző *flag* itt nem egy 8 bites fix bit-kombináció, hanem egy 4 SYN karakterből álló sorozat. A *flag* után a start jelzés következik, amelyet egy DLE STX karakterpár lát el. Ez indítja a ciklikus redundancia számlálót is. A start után a cím karakter és a vezérlő karakter következik, amelyek teljesen megegyeznek felépítés és szerep tekintetében az eredeti leírás cím és vezérlő mezejével. *I-frame* generálása esetén a vezérlő karakter után tetszőleges számú adatkarakter következhet, amelyet egy DLE ETX karakter-kettős zár le. Ez a stop jelzés szerepét tölti be s leállítja a ciklikus redundancia számlálót. Mivel a DLE karakternek speciális szerepe van, ezért a start és a stop jelzés között előforduló ilyen bit-kombinációt az adó oldalon megismétlik, a vevő oldalon pedig az egyiket kiszűrik *(karakter-stuffing technika)*. A stop jelzés után két ciklikus redundancia ellenőrző karakter következik, majd a *flag* karakter-négyes zárja a *frame*-et. *S*- vagy *U-frame* generálásakor nincs adatmező, ezért a vezérlő karakter után közvetlenül a stop jelzés, majd az ellenőrző karakterek következnek.

Összefoglalva az elmondottakat a *frame*-ek szerkezete a következő:

I-frame:



S- és U-frame:



A „párbeszéd" folyamatának dekompozíciója

A „párbeszéd" folyamatának felbontását az előbbiek figyelembevételével az előző fejezetben bemutatott módszertől eltérő módon kell elvégezni. Az alacsonyabb szintű részfolyamatok az alábbiak lesznek:

(1) *Frame-generálási folyamat,* ahol a *frame*-et különböző *részframe*-ekből állítjuk elő.
(2) *Részframe-generálási folyamat,* ahol a *részframe*-ek mezőkből képződnek.
(3) *Mező-generálási folyamat,* ahol a mezőket karakterekből állítjuk elő.
(4) *Karakter-generálási folyamat,* ahol a karaktereket bitekből állítjuk össze.

A hierarchikus rendszer leírása

A *frame-generálás* során a KFKI-*frame*-et részmezőkből állítjuk össze. A generálási folyamatot a G^2 nyelvtan írja le.

$G^2=(\{S^2, A^2, B^2\}, \{flag,$ védett, ellenőrző$\}$,

$\{S^2 \to flag\ A^2;\ A^2 \to$ védett $B^2;\ B^2 \to$ ellenőrző$\}, S^2)$.

A *részframe-generálási folyamat* leírására egy $\mathcal{G}^3$ nyelvtan-halmaz szolgál.

$$\mathcal{G}^3 = \{G_F^3, G_V^3, G_E^3\}, \quad \text{ahol}$$

$$G_F^3 = (\{S_F^3, A_F^3\}, \{\varepsilon\}, \{S_P^3 \to A_P^3\}, S_F^3)$$

a *flag* generálási folyamatot tükrözi;

$G_V^3=(\{S_V^3, A_V^3, B_V^3, C_V^3\}$, {start, cím, vezérlő, adat, stop},

$\{S_V^3 \to$start $A_V^3;\ A_V^3 \to$ cím $B_V^3;\ B_V^3 \to$ vezérlő $C_V^3;$

$C_V^3 \to$ adat $C_V^3;\ C_V^3 \to$ stop$\}, S_V^3)$

a védett részframe generálását írja le;

$$G_E^3 = (\{S_E^3, A_E^3\}, \{\varepsilon\}, \{S_E^3 \to A_E^3\}, S_E^3)$$

az ellenőrző részframe generálását mutatja.

A *mező-generálás* során a mezőket karakterekből állítjuk össze. A mező-generálás a $\mathscr{G}^4$ nyelvtan-halmazzal reprezentálható.

ahol

$$\mathscr{G}^4 = \{G_F^4, G_S^4, G_C^4, G_V^4, G_A^4, G_Z^4, G_E^4\},$$

$$G_F^4 = (\{S_F^4, A_F^4, B_F^4, C_F^4\}, \{\overline{\text{SYN}}, \text{SYN}\}, \{S_F^4 \to \text{SYN}\, A_F^4;$$

$$S_F^4 \to \overline{\text{SYN}}\, S_F^4;\ A_F^4 \to \overline{\text{SYN}}\, B_F^4;\ A_F^4 \to \overline{\text{SYN}}\, S_F^4;$$

$$B_F^4 \to \text{SYN}\, C_F^4;\ B_F^4 \to \overline{\text{SYN}}\, S_F^4;\ C_F^4 \to \text{SYN};\ C_F^4 \to \overline{\text{SYN}}\, S_F^4\},\ S_F^4)$$

a *flag* mező generálását írja le ($\overline{\text{SYN}} = V_T^4 \setminus \text{SYN}$, ahol V_T^4 a teljes karakter-készlet);

$$G_S^4 = (\{S_S^4, A_S^4\}, \{\text{DLE}, \text{STX}, \overline{\text{DLE}}, \overline{\text{STX}}\}, \{S_S^4 \to \text{DLE}\, A_S^4;$$

$$S_S^4 \to \overline{\text{DLE}}\, S_S^4;\ A_S^4 \to \text{STX};\ A_S^4 \to \overline{\text{STX}}\, S_S^4\},\ S_S^4)$$

a start mező generálását írja le ($\overline{\text{DLE}} = V_T^4 \setminus \text{DLE}$; $\overline{\text{STX}} = V_T^4 \setminus \text{STX}$);

$$G_C^4 = (\{S_C^4, A_C^4\}, \{\text{ADR}, \text{DLE}\}, \{S_C^4 \to \text{ADR};\ S_C^4 \to \text{DLE}\, A_C^4;\ A_C^4 \to \text{ADR}\},\ S_C^4)$$

a cím mező generálását tartalmazza ($\text{ADR} \in V_T^4$);

$$G_V^4 = (\{S_V^4, A_V^4\}, \{\text{CTR}, \text{DLE}\}, \{S_V^4 \to \text{CTR};\ S_V^4 \to \text{DLE}\, A_V^4;\ A_V^4 \to \text{DLE}\},\ S_V^4)$$

a vezérlő mező generálását leíró nyelvtan ($\text{CTR} \in V_T^4$);

$$G_A^4 = (\{S_A^4, A_A^4\}, \{\text{CHAR}, \text{DLE}\}, \{S_A^4 \to \text{CHAR}\, S_A^4;\ S_A^4 \to \text{CHAR};$$

$$S_A^4 \to \text{DLE}\, A_A^4;\ A_A^4 \to \text{CHAR}\, S_A^4;\ A_A^4 \to \text{DLE}\},\ S_A^4)$$

az adat-mező generálását szemléltető nyelvtan ($\text{CHAR} \in V_T^4$);

$$G_Z^4 = (\{S_Z^4, A_Z^4\}, \{\text{DLE}, \text{ETX}, \overline{\text{DLE}}, \overline{\text{ETX}}\}, \{S_Z^4 \to \text{DLE}\, A_Z^4;$$

$$S_Z^4 \to \overline{\text{DLE}}\, S_Z^4;\ A_Z^4 \to \text{ETX};\ A_Z^4 \to \overline{\text{ETX}}\, S_Z^4\},\ S_Z^4)$$

a stop mező leírására szolgáló nyelvtan ($\overline{\text{ETX}} = V_T^4 \setminus \text{ETX}$);

$$G_E^4 = (\{S_E^4, A_E^4\}, \{\text{CRC1}, \text{CRC2}\}, \{S_E^4 \to \text{CRC1}\, A_E^4;\ A_E^4 \to \text{CRC2}\},\ S_E^4)$$

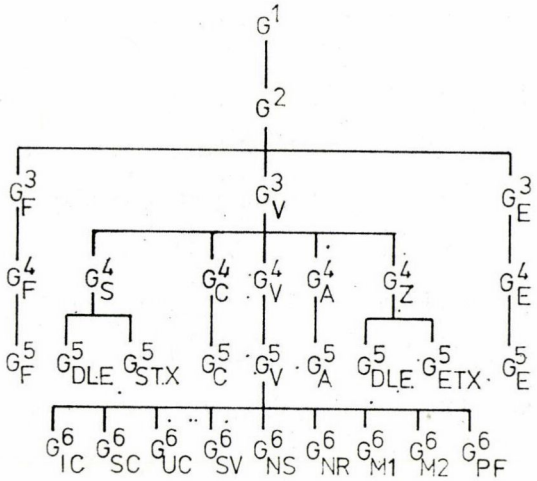
az ellenőrző mező generálását tükrözi.

A *karakter-generálási* folyamat során a karaktereket bitekből állítjuk össze az alkalmazott kód-készlet típusának megfelelően (pl. ASCII, EBCDIC stb.). Egyedül a vezérlő mezőt reprezentáló CTR karakter generálásáról kell külön szólnunk.

A CTR karakter belső szerkezete megegyezik a HDLC eljárás vezérlő mezejének szerkezetével, tehát ez a karakter további részegységekre bontva, még egy további szintet bevezetve képezhető hasonló módon, mint ahogy azt az előző fejezetben a vezérlő mező generálásakor bemutattuk.

A szintek közötti *interface* az előző fejezetben bemutatott módszer alapján definiálható itt is.

A *rendszer-nyelvtan* struktúráját a 3. ábra mutatja.



3. ábra

7. A bemutatott módszer értékelése

A dolgozatban bemutatott módszer adatátviteli vagy számítógéphálózatok protokolljainak tervezéséhez és implementálásához nyújt segítséget. Alkalmazásával lehetővé válik az, hogy más digitális rendszerhez hasonlóan, a protokollokat is formális leírásra alkalmas eszközökkel definiáljuk. Az itt használt eszköz, a formális nyelvtan, megfelelőnek bizonyult mind a HDLC-protokoll, mind az általunk módosított változata struktúrájának formális leírására.

A bemutatott módszer segítségével egy protokoll szintet definiáltunk formálisan úgy, hogy egy rendszer-nyelvtant generáltunk, amely nagyvonalaiban is és részleteiben is pontosan követi a szöveges leírásban meghatározott szabályokat, illetőleg a formális definíció alapján a szöveges leírás pontosan és egyértelműen rekonstruálható. A módszer segítséget nyújt az implementáláshoz azáltal, hogy a „párbeszéd" folyamatának többszintű felbontásával tulajdonképpen a strukturált programozás elveinek megfelelő önálló modulok képződtek, amelyek megfelelő egymás fölé/alá rakásával, a definiált *interface*-en keresztül a teljes protokoll szint *software* rendszere felépíthető.

Mindezek ellenére azonban további munka szükséges ahhoz, hogy ezzel vagy más megoldással a protokoll szementikáját formálisan is definiálni tudjuk. További vizsgálatot igényel a protokoll-hierarchia formális leírása is.

IRODALOM

[1] MADNICK, S. E. and DONOVAN, J. J., *Operating Systems* (McGraw-Hill, New York, 1974).
[2] *High Level Data Link Control Procedures. Proposed Draft International Standard on Elements of Procedures* (ISO/TC/SC6, N—1005, 1975).
[3] *High Level Data Link Control Procedures. Frame Structures* (ISO DIS—3309.2, 1975).
[4] CERF, V. G., "Multiprocessors, Semaphores and a Graph Model of Computation", Ph. D. Dissertation, UCLA—ENG—7223, 1972.

[5] POSTEL, J. B., "A Graph Model Analysis of Computer Communications Protocols", Ph. D. Dissertation, UCLA—ENG—7410, 1974.
[6] BJORNER, D., "Finite state automation — Definition of data communication line control procedures", *AFIPS Conference Proceedings* **37** (1970) 477—491.
[7] RUSBRIDGE, R. E. and LANGSFORD, A., *Formal Representation of Protocols for Computer Networks* (AERE—R 7826, 1974).
[8] MERLIN, P. M. and FARBER, D. J., "Recoverability of communication protocols — implications of a theoretical study", *IEEE Transactions on Communications* **COM—24** (1976) 1036—1043.
[9] HOFFMANN, H. J., *On Linguistic Aspects of Communication Line Control Procedures* (IBM-Report, RZ 345, 1970).
[10] HEART, F. E. et al., "The interface message processor for the ARPA computer network", *AFIPS Conference Proceedings, SJCC 1970* **36** (1970) 551—567.
[11] *To the participants of the IIASA CSN from the Hungarian party on the LINE CONTROL PROCEDURE* (IIASA CSN 019, 1976).

(Beérkezett: 1977. március 29.)

HARANGOZÓ JÓZSEF
MTA KÖZPONTI FIZIKAI KUTATÓ INTÉZET
1525 BUDAPEST PF. 49.

FORMAL DEFINITION FOR A DATA LINK LEVEL PROTOCOL OF COMPUTER NETWORKS

J. HARANGOZÓ

The communication protocols of the data communication and the computer networks form a hierarchical multilevel structure. This paper deals with the logical analysis of the data link level protocol of this structure. It attempts to give a formal description for the subset of a data link level protocol recommended by ISO — the *High Level Data Link Control Procedure* — with a formal language.

The interaction between two element of the network during the data link is implemented by different information, supervisory and unnumbered frames, which can be considered symbols and the interaction can be considered a string of symbols. The strings are elements of a definite formal language, which — by BJORNER and HOFFMANN — may be defined by a regular grammar. The paper gives a method to describe this protocol with a definite regular grammar. Further regular grammars for formal description of lower layers of operation of the "dialogue" level can be constructed by decomposition. A same grammar can be given for an ARPA-like link level protocol, which will be implemented in a local computer network to be developed in the Central Research Institute for Physics.

Alkalmazott Matematikai Lapok **2** *(1976) 331—340*

SZIMMETRIKUS SŰRŰSÉGFÜGGVÉNYEK SZUPERPOZÍCIÓINAK FELBONTÁSÁRÓL[1]

MEDGYESSY PÁL†
Budapest

1.

Legyen $f(x)$ adott analitikus alakú, (0) *szimmetrikus, szigorúan* (0) *egycsúcsú,* folytonos sűrűségfüggvény[2]. A

$$k(x) = \sum_{k=1}^{N} p_k f\left(\frac{x-\alpha_k}{\beta_k}\right)$$

függvényt, ahol $p_k>0$, azonos (α_k, β_k) párok nincsenek és $0<\beta_1\leqq\beta_2\leqq\ldots\leqq\beta_N$, az $f((x-\alpha_k)/\beta_k)$ sűrűségfüggvények — *a komponensek* — p_k *súlyokkal* képzett *szuperpozíciójának* nevezzük.

A gyakorlatban sokszor találkozunk a következő *problémával.*

Adva vannak egy $k(x)$ *szimmetrikus sűrűségfüggvény-szuperpozíció* görbéje egyes pontjai ordinátáinak *mért* értékei. Az N, p_k, α_k, β_k paraméterek ismeretlenek. Meghatározandó a *mért* értékek alapján N és esetleg egyes p_k, α_k, β_k paraméterek *közelítő értéke.*

A problémánk megoldását szolgáltató *numerikus* eljárást a $k(x)$ *sűrűségfüggvény-szuperpozíció* (numerikus) *felbontásának* nevezzük.

Részletes tárgyalás és példák: [1], I. 1. §.

Legyen $f_1(x)$, illetve $f_2(x)$ (0) szimmetrikus, szigorúan (0) egycsúcsú sűrűségfüggvény. Azt mondjuk, hogy $f_2(x)$ görbéje (tágabb értelemben) *keskenyebb*, mint $f_1(x)$ görbéje, ha 1. $x>0$ esetén az $f_2(x)$ sűrűségfüggvényhez tartozó eloszlásfüggvény görbéje az $f_1(x)$ sűrűségfüggvényhez tartozó eloszlásfüggvény görbéje felett halad és 2. $f_2(x)$ görbéjének csúcsmagassága nagyobb, mint $f_1(x)$ görbéjének csúcsmagassága.

Ekkor $f_2(x)$ görbéje $f_1(x)$ görbéjéből abszcisszatengely menti összenyomás és ordinátatengely menti nyújtással keletkezik, miközben a görbe alatti terület változatlan.

Megállapodunk abban, hogy $f_1(x)$ és $f_2(x)$ eltolása nem változtat a „keskenyebb" viszonylaton.

Tekintsünk most egy $\{g(x, \lambda)\}$ $(\Lambda_1\leqq\lambda<\Lambda_2)$ egyparaméteres sűrűségfüggvénycsaládot, melynek tagjai (0) szimmetrikusak és szigorúan (0) egycsúcsúak. Ha λ_i, λ_j a λ paraméter két különböző értéke, $\Lambda_1<\lambda_i<\lambda_j<\Lambda_2$ és $g(x, \lambda_i)$ görbéje (tágabb

[1] Kiegészítések a [3] dolgozathoz.

[2] Ha $-\infty<x<\infty$, $f(x)$ „(a) szimmetrikus" azt jelenti, hogy $f(a-x)=f(a+x)$; $f(x)$ „szigorúan (a) egycsúcsú" pedig azt, hogy $f(x)$ szigorúan egycsúcsú, $x=a$ csúcshellyel.

értelemben) keskenyebb, mint $g(x, \lambda_i)$ görbéje, azt mondjuk, hogy λ $g(x, \lambda)$ görbéjének (Λ_1, Λ_2) *monoton formánsa.* (Λ_1, Λ_2) λ monoton változása irányára utal. — E monoton formáns szemléletes jelentése egyszerű: ha λ monoton növekedik, a megfelelő görbe egyre keskenyebb lesz; a keskenységet egyetlen szám jellemzi.

Részletek és egyéb „keskenyebbség" — definíciók: [1], II. 4. §.

A fenti $k(x)$ szuperpozíció esetére a felbontási eljárások lényege többek között a következőképp fogalmazható meg ([2], III. 1. §.).

A görbéjével képviselt $k(x)$ szuperpozícióhoz hozzárendelünk egy

$$b(y) = \sum_{k=1}^{N} p_k g_1(y, \alpha_k, \beta_k)$$

típusú úgynevezett *tesztfüggvényt*, melyre fennáll:

1. $g_1(y, \alpha_k, \beta_k)$ szigorúan egycsúcsú sűrűségfüggvény, $f\left(\frac{x-\alpha_k}{\beta_k}\right)$ és $g_1(y, \alpha_k, \beta_k)$ között jól definiált összefüggés áll fenn; emellett $g_1(y, \alpha_k, \beta_k)$ görbéje keskenyebb, mint $f\left(\frac{x-\alpha_k}{\beta_k}\right)$ görbéje;

2. ha $(\alpha_k, \beta_k) \not\equiv (\alpha_l, \beta_l)$, akkor $g_1(y, \alpha_k, \beta_k)$ görbéjének csúcshelye $g_1(y, \alpha_l, \beta_l)$ görbéjének csúcshelyétől legalább olyan távol van, mint $f\left(\frac{x-\alpha_k}{\beta_k}\right)$ görbéjének csúcshelye $f\left(\frac{x-\alpha_l}{\beta_l}\right)$ görbéjének csúcshelyétől;

3. $k(x)$ és $b(y)$ között összefüggés állítható fel az $f\left(\frac{x-\alpha_k}{\beta_k}\right)$ és $g_1(y, \alpha_k, \beta_k)$ közötti összefüggés alapján.

Mindebből az következik, hogy ha a $b(y)$ *sűrűségfüggvény-szuperpozíció* görbét felrajzolnánk, abban az egyes komponensek görbéi *különváltabban* mutatkoznának meg, mint $k(x)$ görbéjében. Ha e különváltság elég nagymérvű, a komponensek görbéi szinte egymást nem is zavarva jelennének meg. Ekkor $b(y)$ görbéjéből a komponensek száma — és esetleg egyes paraméterek közelítő értéke is — megállapítható volna.

Mivel mindez *közelítőleg* igaz akkor is, ha $k(x)$ görbéjének *mért* ordinátaértékeiből $b(y)$ analitikus alakjának megfelelő numerikus módszerrel a $b(y)$ tesztfüggvény bizonyos közelítésének görbéjét állítjuk elő, *felbontási eljárásnak ez utóbbi görbe csúcsai számának, helyeinek stb. megállapítását fogjuk tekinteni.*

Adott felbontási probléma esetében a feladat tehát:

A) az említett $b(y)$ tesztfüggvény megtalálása;

B) a felbontandó szuperpozíció komponensei és a tesztfüggvény komponensei közötti összefüggés meghatározása, és ennek alapján a felbontási eljárás alapjául szolgáló $b(y)$ tesztfüggvény előállítása a $k(x)$ szuperpozíció segítségével;

C) mindezek alapján numerikus módszer kidolgozása a tesztfüggvény valamilyen közelítése görbéjének előállítására.

Részletek és általánosítás: [2], III. 1. §.

2.

Korábbi munkáinkban a $b(y)$ tesztfüggvényt a $k(x)$ szuperpozíciót tartalmazó konvolúciós integrálegyenlet megoldása vagy konvolúciós transzformált szolgáltatta. A tesztfüggvénynek az előbbiekre épített analitikus kifejezését a *numerikus felbontáskor* egy

$$b^*(y) = \sum_{j=-m}^{m} c_j k(x+jh)$$

alakú összeggel *közelítettük* (h adott konstans).

E közelítés *hibája* azonban nehezen volt becsülhető; a kapott becslések a gyakorlatban használhatatlanok is voltak. $b^*(y)$ görbéjében általában számos olyan kisebb csúcs mutatkozott, amelyről nem tudtuk megmondani, hogy $b(y)$ valamelyik komponense közelítésének görbéje-e vagy pedig a $k(x)$ mért értékeiben rejlő hibáktól, „zajtól" származik-e. A gyakorlatban előadódó szuperpozíciók felbontását ez sokszor megnehezítette.

A címben idézett [3] *dolgozatunk egy új felbontási módszert ismertetett, mely kiküszöböli az említett nehézségeket.* E módszert ott két egyszerű példán mutattuk be. Most egy sokkal bonyolultabb példát fogunk látni, előbb azonban felelevenítjük a módszer alapötletét.

Új módszerünk *alapötlete:* adott $k(x)$ szuperpozícióhoz tesztfüggvényként $p(y)=\sum_{\nu=-m}^{m} c_\nu k(x+\nu h)$ típusú *véges* összeget kísérelünk meg hozzárendelni, minél kisebb m érték mellett ($m=1$ vagy 2), a c_ν konstansokat a tesztfüggvényt értelmező feltételekből határozva meg. Ez esetben numerikus felbontáskor a tesztfüggvény analitikus kifejezésének közelítése elmarad és a $k(y)$ mért értékeiben rejlő „zajnak" a tesztfüggvényt eltorzító hatása is követhető. A $p(y)$ tesztfüggvény bevezetése tehát a keresett módszert szolgáltatja.

Hogy vizsgálódásunkat folytathassuk, határozottabb formát kell adnunk alapötletünknek. A később tárgyalandó példákat tartva szem előtt, $p(y)$ konkrét alakja a következő meggondolások alapján adható meg:

Legyen $f(x)$ a fenti (0) szimmetrikus, szigorúan (0) egycsúcsú sűrűségfüggvény és tekintsük a

$$h(y, P, \vartheta) = (1+2P)f(y)-P[f(y-\vartheta)+f(y+\vartheta)]$$

függvényt, ahol $P\geqq 0$, $\vartheta\geqq 0$. Látható, hogy $h(y, P, \vartheta)$ is (0) szimmetrikus és $h(y, 0, \vartheta)=f(y)$.

Tegyük fel, hogy bármely $\vartheta>0$-hoz található egy a ϑ-tól függő $P_0(\vartheta)$ korlát, melyre fennáll:

A) ha $0\leqq P<P_0(\vartheta)$, $h(y, P, \vartheta)$ nemnegatív, mikoris $h(y, P, \vartheta)$ sűrűségfüggvény;

B) ha $0\leqq P<P_0(\vartheta)$, $h(y, P, \vartheta)$ szigorúan (0) egycsúcsú, vagyis $h'_y(y, P, \vartheta)\leqq 0$ ($y>0$) és nincs olyan nem félig végtelen intervallum az abszcisszatengelyen, melynek minden pontjában $h'_y(y, P, \vartheta)=0$.

C) ha $0\leqq P<P_0(\vartheta)$, P $h(y, P, \vartheta)$ görbéjének $(0, P_0(\vartheta))$ monoton formánsa, vagyis P növekedésekor $h(y, P, \vartheta)$ görbéje egyre keskenyebbé válik, olyan értelemben, hogy (az origóban levő) csúcsa emelkedése mellett (ami rögtön látható), a görbe az y-tengely felé irányulva összenyomódik.

Ezek után vegyük a $k(x)=\sum_{k=1}^{N} p_k f\left(\frac{x-\alpha_k}{\beta_k}\right)$ szuperpozícióhoz rendelt tesztfüggvénynek a következő $p(y,\lambda)=\sum_{k=1}^{N} p_k g_1(y,\alpha_k,\beta_k)$ függvényt, melyben egy később meghatározandó értékkészletű λ paraméter is szerepel és ϑ rögzített:

$$p(y,\lambda)=(1+2\lambda)k(x)-\lambda[k(x-\vartheta)+k(x+\vartheta)].$$

$k(x)$ k-adik komponense, $f\left(\frac{x-\alpha_k}{\beta_k}\right)$ és $p(y,\lambda)$ k-adik komponense, $g_1(y,\alpha_k,\beta_k)$ között a

$$g_1(y,\alpha_k,\beta_k)=(1+2\lambda)f\left(\frac{y-\alpha_k}{\beta_k}\right)-\lambda\left[f\frac{y-\vartheta-\alpha_k}{\beta_k}\right)+f\left(\frac{y+\vartheta-\alpha_k}{\beta_k}\right)\Big]$$

összefüggés áll fenn; mivel ez λ-t is tartalmazza, de csak $y-\alpha_k$-tól függ, vezessük be a $g_1(y,\alpha_k,\beta_k)=g(y-\alpha_k,\beta_k,\lambda)$ jelölést; ezzel

$$p(y,\lambda)=\sum_{k=1}^{N} p_k g(y-\alpha_k,\beta_k,\lambda).$$

Mivel skálaparaméter-változtatás vagy eltolás nemnegativitáson, egycsúcsúságon, keskenységi viszonylaton nem változtat, az előbbiek folytán $g(y-\alpha_k,\beta_k,\lambda)$ (α_k) szimmetrikus, szigorúan (α_k) egycsúcsú sűrűségfüggvény és görbéjének λ $\left(0, P_0\left(\frac{\vartheta}{\beta_k}\right)\right)$ monoton formánsa. Így tehát görbéje keskenyebb, mint $f\left(\frac{x-\alpha_k}{\beta_k}\right)=$ $=g(x-\alpha_k,\beta_k,0)$ görbéje, ha $0<\lambda<P_0\left(\frac{\vartheta}{\beta_k}\right)$ és λ növekedésével keskenysége növekedik. Továbbá $g(y-\alpha_k,\beta_k,\lambda)$ és $g(y-\alpha_l,\beta_l,\lambda)$ görbéi csúcshelyeinek távolsága ugyanakkora, mint $f\left(\frac{x-\alpha_k}{\beta_k}\right)$ és $f\left(\frac{x-\alpha_l}{\beta_l}\right)$ görbéi csúcshelyeinek távolsága (azaz $|\alpha_k-\alpha_l|$).

Így tehát a $p(y,\lambda)$ és $k(x)$ közti fenti összefüggés tesztfüggvényt szolgáltat, ha λ értékére fennáll $0<\lambda<P_0\left(\frac{\vartheta}{\beta_k}\right)$ $(k=1,\ldots,N)$, vagyis mindig, amidőn $0<\lambda<P_0\left(\frac{\vartheta}{\beta_N}\right)$. Minél közelebb van λ $P_0\left(\frac{\vartheta}{\beta_N}\right)$-hez, annál keskenyebbé válnak a tesztfüggvény komponenseinek görbéi, elsősorban a β_N paraméterű és annál inkább sikerülhet a felbontás. A legjobb komponens-különválást nyújtó λ értékéből $P_0\left(\frac{\vartheta}{\beta_N}\right)$-re, vagyis β_N-re következtethetünk; ennél nagyobb λ alkalmazásakor viszont negatív tesztfüggvényértékek léphetnek fel és a tesztfüggvény görbéje áttekinthetetlenné válik.

A gyakorlatban az a helyzet, hogy ϑ megválasztása után (amit az is befolyásol, hogy milyen pontokban adott $k(x)$ mért értéke) egyre növekedő λ értékekkel előállítjuk $p(y,\lambda)$ görbéjét, $k(x)$ mért értékei segítségével. Ezek a tesztfüggvény bizonyos közelítésének görbéi lesznek, melyekben annál jobban várható különálló komponens görbék megmutatkozása (csúcsok fellépése), minél közelebb van az épp használt λ $P_0\left(\frac{\vartheta}{\beta_N}\right)$-hez.

Ami ϑ megválasztását illeti, mindeddig abból indultunk ki, hogy $\vartheta(\vartheta>0)$ tetszőleges, illetve, hogy $k(x)$ rendelkezésre álló mért értékei jelölik ki a szóbajövő ϑ-értékeket. A $\lambda\uparrow P_0\left(\frac{\vartheta}{\beta_N}\right)$ esetében fellépő keskenyebbé válás mértéke azonban ϑ-tól függ. A keskenyebbé válás szempontjából legjelentősebb a $g(y-\alpha_k, \beta_k, \lambda)$ $(k=1, \ldots, N)$ komponenseinek görbéi csúcsmagasságának megnövekedése, feltéve, hogy λ megnövekedésekor e görbe „monoton" esik, emellett az y-tengely felé irányulóan összenyomódik. A csúcsmagasságok nyilván

$$g(0, \beta_k, \lambda) = (1+2\lambda)f(0)-2\lambda f\left(\frac{\vartheta}{\beta_k}\right) \quad (k=1, \ldots, N);$$

$\lambda\uparrow P_0\left(\frac{\vartheta}{\beta_N}\right)$ esetén ennek minél gyorsabban növekednie kell. ϑ és $P_0(\vartheta)$ ismeretében az optimális λ-érték *elvileg* megtalálható. Vegyük azonban figyelembe, hogy a gyakorlatban

$$\hat{f}\left(\frac{x-\alpha_k}{\beta_k}\right) = f\left(\frac{x-\alpha_k}{\beta_k}\right)+\varepsilon_k(x)$$

áll rendelkezésünkre, ahol $\varepsilon_k(x)$ a „zaj" és e „zaj" továbbplántálódik $g(y-\alpha_k, \beta_k, \lambda)$ kiszámított értékeibe is, azokat $\hat{g}(y-\alpha_k, \beta_k, \lambda)=g(y-\alpha_k, \beta_k, \lambda)+\xi_k(y)$-ná torzítva el. Könnyen belátható, hogy ha $|\varepsilon_k(x)|<E$, a $\xi_k(y)$ eltorzítása fennáll $|\xi_k(y)|<$ $<(1+4\lambda)E$, vagyis ez az eltorzítás nagyjából $(1+4\lambda)$-val arányosan nő; λ azonban ϑ függvénye. Hasonlók igazak $p(y, \lambda)$-nak $k(x)$ „zaja" okozta eltorzulására. A „zaj" megnövekedése miatt nem ajánlatos tehát olyan ϑ-t választani, amely mellett λ nagy lesz. — Világos, hogy a „zaj", mint sztochasztikus folyamat-realizáció továbbplántálódása is elemi eszközökkel vizsgálható; erre itt nem térünk ki.

A gyakorlatban legjobb több, monoton változó ϑ értékkel kísérletezni, és a kapott tesztfüggvény-görbéket egybevetve keresni ki a legjobb felbontást mutatókat.

$P_0(\vartheta)$ értékét adott típusú $f\left(\frac{x-\alpha_k}{\beta_k}\right)$ komponensekből álló $k(x)$ szuperpozíció esetében analitikus eszközökkel vagy numerikus kísérletezéssel — $h(y, P, \vartheta)$ értékeit sokféle (P, ϑ) párra tabellázva — határozzuk meg. Ha $P_0(\vartheta)$ egyszer tabellázva van, e táblázat $f(x)$ típusához tartozó univerzális konstans-összesség lesz.

Belátható, hogy — korábbi módszereinkkel ellentétben — a közölt eljárás általában *nem* biztosítja azt, hogy a $p(y, \lambda)$ tesztfüggvénynek optimális esetben legalább egy komponense tetszőlegesen keskeny (végtelenbe futó csúcshoz hasonló) görbéjű lesz. Ez például abból is következik, hogy rögzített ϑ mellett a két, megengedett P_1 és P_2 $(P_2>P_1)$ paraméterértékhez tartozó $h(y, P_1, \vartheta)$ és $h(y, P_2, \vartheta)$ függvények görbéi metszéspontjának abszcisszája csupán ϑ-tól függ és általában nem tart zérushoz, még ha P_2-höz keskenyebb görbe tartozik is, mint P_1-hez. Nevezetesen az

$$(1+2P_1)f(y)-P_1[f(y-\vartheta)+f(y+\vartheta)] =$$

$$= (1+2P_2)f(y)-P_2[f(y-\vartheta)+f(y+\vartheta)]$$

összefüggésből

$$2f(y)-[f(y-\vartheta)+f(y+\vartheta)] = 0$$

következik, vagyis *nagyjából* $f''(y)=0$, és az ezen egyenlet megoldását szolgáltató $y=y_0$ érték általában nem 0, vagy ∞, vagyis P növekedésekor $h(y, P, \vartheta)$ görbéje nem közeledik egyre jobban és jobban az ordinátatengelyhez.

Konkrét esetben tehát lehetséges, hogy új eljárásunk nem nyújtja a kívánt felbontást; ezt azonban csak kísérletezés döntheti el. Kétségtelen előnye mindenesetre egyszerűsége és a hibák (a „zaj") befolyásának áttekinthető volta.

Mindezekre tekintettel új eljárásunkat a $k(x)$ szuperpozíció részleges, *parciális* (numerikus) *felbontásának* nevezzük.

Alapgondolata akkor is alkalmazható, ha $p(y, \lambda)$ helyett ennél bonyolultabb, $\sum_{\nu=-m}^{m} c_\nu k(x+\nu h)$ $(m=2, 3, \ldots)$ alakú tesztfüggvényt vezetünk be. Részletekbe azonban itt nem bocsátkozhatunk.

1. *Megjegyzés.* A fentiekből következik, hogy ha $f(x)$ karakterisztikus függvénye $\varphi(t)$, akkor λ és ϑ minden olyan értékére, amelyre $p(y, \lambda)$ tesztfüggvény, $p(y, \lambda)$ *Fourier-transzformáltja,* $\psi(t)=\varphi(t)\left[1+4\lambda \sin\frac{2\vartheta t}{2}\right]$ *karakterisztikus függvény.*

4.

[3] 4. szakaszában a leírt módszert két példán mutattuk be: *Laplace-sűrűségfüggvények, valamint ch-sűrűségfüggvények szuperpozíciójának felbontásán.*

Ezekben a (0) szimmetrikus, szigorúan (0) egycsúcsú $f(x)$ sűrűségfüggvénnyel megalkotott

$$k(x)=\sum_{k=1}^{N} p_k \frac{1}{\beta_k} f\left(\frac{x-\alpha_k}{\beta_k}\right) \quad (p_k>0,\ \alpha_k \neq \alpha_l\ (k \neq l),\ 0<\beta_1 \leqq \ldots \leqq \beta_N)$$

sűrűségfüggvény-szuperpozíció és az annak felbontására felhasználható

$$p(y, \lambda)=(1+2\lambda)k(x)-\lambda[k(x+\vartheta)+k(x-\vartheta)] \quad \left(\vartheta>0,\ 0<\lambda<P_0\left(\frac{\vartheta}{\beta_N}\right)\right)$$

tesztfüggvény λ és ϑ paraméterei között kapcsolatot létesítő $P_0(\vartheta)$ függvényt explicit alakban meg tudtuk adni.

Sok esetben azonban a $P_0(\vartheta)$ függvényt csak *numerikus kísérletezéssel,* tabellázva tudjuk megadni. Erre világít rá a következő példa.

PÉLDA. *Cauchy-sűrűségfüggvények szuperpozíciójának felbontása.* Legyen

$$k(x)=\sum_{k=1}^{N} p_k \frac{1}{\pi\beta_k} \frac{1}{1+(x-\alpha_k)^2/\beta_k^2}$$

$$(p_k>0,\ \alpha_k \neq \alpha_l\ (k \neq l),\ 0<\beta_1 \leqq \ldots \leqq \beta_N < \Lambda_2).$$

Itt, konstans faktortól eltekintve,

$$h(y, P, \vartheta)=(1+2P)\frac{1}{1+y^2}-P\left[\frac{1}{1+(y+\vartheta)^2}+\frac{1}{1+(y-\vartheta)^2}\right]$$

(elegendő $y \geqq 0$ esetére vizsgálni ezt a függvényt). A $P_0(\vartheta)$ függvényt itt csak *numerikus kísérletezéssel*, tabellázva kaphattuk meg. Közelebbről: különböző ϑ értékekhez meghatároztuk azt a maximális $P = P_{\max}$ értéket, amely mellett $h(y, P, \vartheta)$ még éppen egycsúcsú sűrűségfüggvény maradt bizonyos elég nagy $(0, L)$ intervallumon, amelynek hosszát az szabta meg, hogy az éppen használt ϑ és P értékek és tetszőleges $y > L$ mellett $h(y, P, \vartheta)$ és $h'_y(y, P, \vartheta) < 0$; ez utóbbit a $h'_y(y, P, \vartheta)$ derivált fáradságos, de gépies kiszámításával és eredményül kapott racionális törtfüggvény számlálójának — egy polinomnak — klasszikus algebrai eszközökkel való megvizsgálásával dönthetjük el. $h(y, P, \vartheta) > 0$ eldöntése hasonlóan megy. Természetesen azt is numerikusan ellenőrizni kell, hogy rögzített ϑ és a $P < P_0(\vartheta)$ feltételnek eleget tevő P-értékek növekedő sorozata mellett $P\,h(y, P, \vartheta)$ görbéjének $(0, P_0(\vartheta))$ monoton formánsa lesz-e, vagy egyre „keskenyebb" és magasabb lesz-e $h(y, P, \vartheta)$ görbéje, ahogy P növekedik.

A kísérletezésbe bevont (ϑ, P) értékpárok közül egyesekkel nagyobb, másokkal kisebb „elkeskenyedését" kaptuk $h(y, P, \vartheta)$ görbéjének. Az optimális esetet $\vartheta = 0{,}31$-nél találtuk, mikor is P értékét 0-tól $P_{\max} = 7{,}6$-ig monoton növelve egyre „keskenyebb" egycsúcsú sűrűségfüggvény-görbét nyertünk; $P > 7{,}6$ esetén a görbepontok ordinátái pozitívak maradtak, de az egycsúcsúság megszűnt; így tehát $P_0(0{,}31) = 7{,}6$ vehető. A görbepontok ordinátáit a $(0; 3{,}6)$ intervallumon számítottuk ki; az említett algebrai eszközökkel igazolható volt, hogy a szóban forgó (ϑ, P) értékpárok mellett $y > 3{,}6$ esetében (sőt már kisebb y-értékekre is)

$$h(y, P, \vartheta) > 0 \quad \text{és} \quad h'_y(y, P, \vartheta) < 0$$

(azt is láttuk, hogy $P\,h(y, P, \vartheta)$ görbéjének $(0; 7{,}6)$ monoton formánsa.).

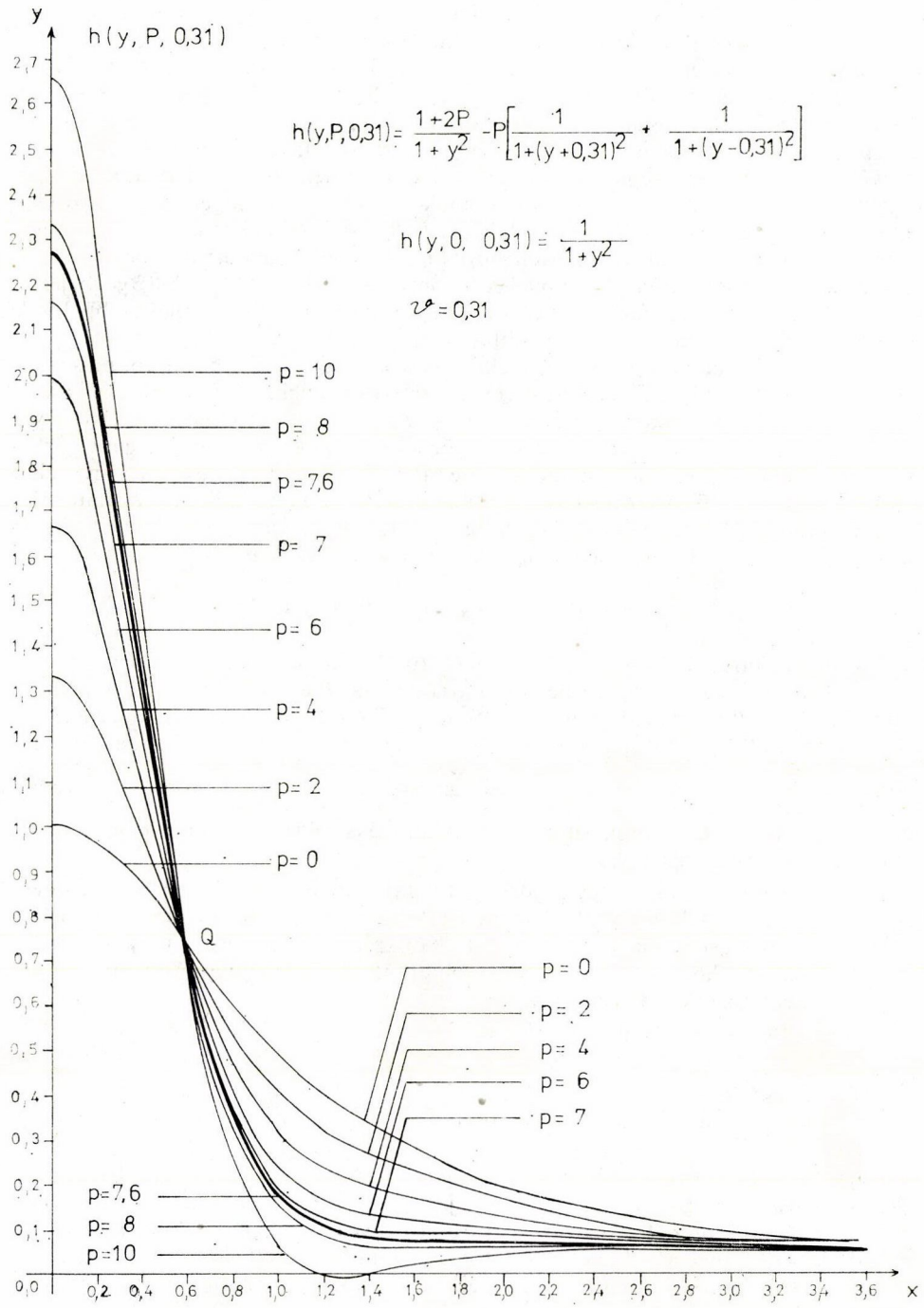
Az 1. ábra $h(y, P, 0{,}31)$ görbéjét mutatja néhány P érték esetére, $y \geqq 0$ mellett. Látható, hogy $P = 8$ is túl nagy, $P = 10$-nél pedig már negatív ordinátaértékek is előfordulnak. $P = 7{,}6$-nál a görbe csúcsmagassága már több, mint kétszerese a kiindulási $f(x) = \dfrac{1}{1+x^2}$ görbe csúcsmagasságának. A $h(y, P, 0{,}31)$-görbék nagyjából mind átmennek a Q ponton (ilyen eset előadódásáról fentebb már volt szó); ez mindenesetre korlátozza „keskenyedésüket".

P aránylag nem nagy és így a görbeordináták „mérési" hibái nem befolyásolják nagyon a parciális felbontásnak a mondottak alapján $k(x)$-re alkalmazható módszerét, a tesztfüggvényt nem torzítják el. Konkrét esetben célszerű mindenesetre nem rögtön $P = 7{,}6$-tal kezdeni el a felbontást. A tesztfüggvénykomponensek itt sem keskenyíthetők el tetszőlegesen.

Metodológiai példaként tekintsük a

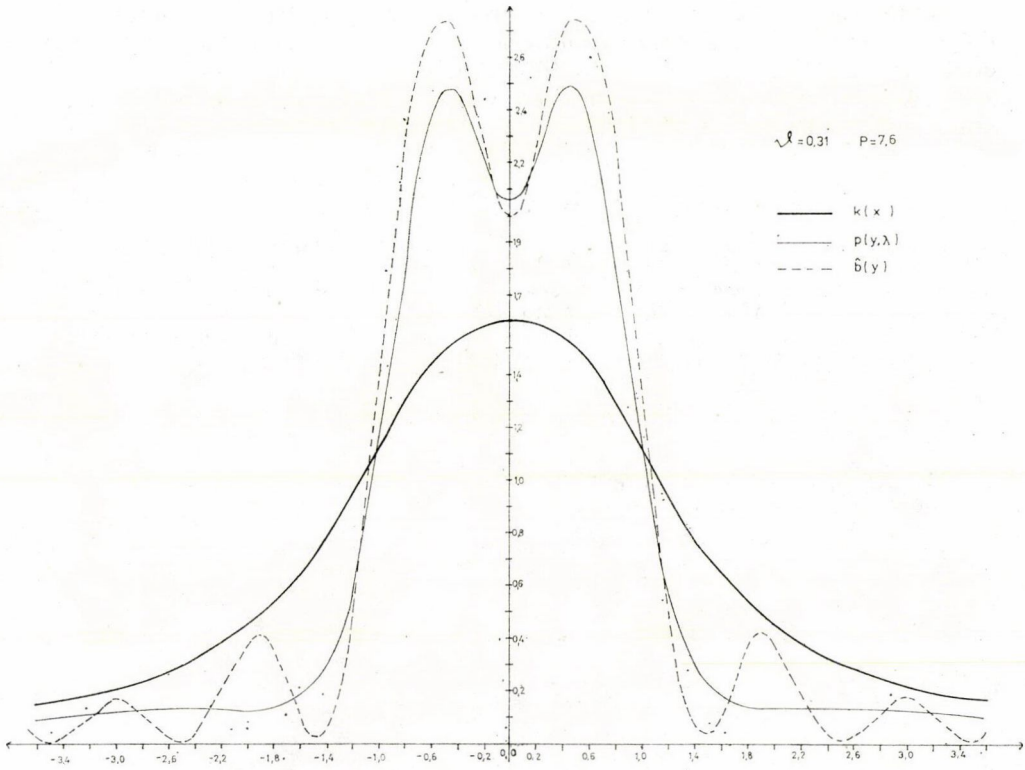
$$k(x) = \frac{1}{1+(x-0{,}5)^2} + \frac{1}{1+(x+0{,}5)^2}$$

Cauchy-sűrűségfüggvény szuperpozíció felbontását. Ezt — más-más módszerrel — [2]-ben többször is elvégeztük. $k(x)$ görbéje egycsúcsú; a 2. ábrán a vastag vonal ábrázolja. A $k(x)$-hez tartozó $p(y, \lambda)$ tesztfüggvény értékeit $\vartheta = 0{,}31$ és $\lambda = P_0(0{,}31) = 7{,}6$ mellett, $k(x)$ táblázatból kiszámított értékeiből 4 tizedesjegyet megtartva számítottuk ki. $p(y, \lambda)$ görbéjét a 2. ábrán a vékony vonal mutatja.



1. ábra

Alkalmazott Matematikai Lapok 2 (1976)



2. ábra

Összehasonlításul bemutatjuk a [2] III. 1. §. 1.1.1.-ben leírt eljárással kapott $\hat{b}(y)$ tesztfüggvény-közelítés görbéjét is (szaggatott vonal); ez több mellék-csúcsot mutat, szemben most nyert *egycsúcsú* tesztfüggvény-görbénkkel. A két — eredetileg rejtett — komponens már $p(y, \lambda)$ görbéjében is jól megmutatkozik.

1. Megjegyzés. Normális sűrűségfüggvények szuperpozíciója felbontásakor a bemutatott eljárás — sajnos — teljesen csődöt mond. Ez rögtön látható abból, hogy a

$$h(y, P, \vartheta) = (1+2P)e^{-\frac{y^2}{4}} - P\left[e^{-\frac{(y+\vartheta)^2}{4}} + e^{-\frac{(y-\vartheta)^2}{4}}\right]$$

függvény deriváltjának *bármely* (ϑ, P) $(\vartheta > 0,\ P > 0)$ értékpár mellett előjelváltása van. A numerikus kísérletezés azt mutatja, hogy módszerünk értelemszerű általánosítása (például $\sum_{j=-2}^{2} c_k k(x+j\vartheta)$ típusú tesztfüggvény bevezetése) sem jár eredménnyel.

2. Megjegyzés. Megoldatlan *probléma*, általában milyen típusú $k(x)$ sűrűségfüggvények esetében alkalmazható *eleve* eljárásunk.

3. Megjegyzés. Cauchy-sűrűségfüggvények szuperpozíciójának felbontására kidolgozott fenti módszerünk *megalapozás nélkül*, csupán tapasztalatilag nyert (ϑ, P) értékpárokkal elvétve eddig is előfordult a fizikában.

Köszönetünket fejezzük ki DELLAGRAMMATICA KULÁnak az 1. és 2. ábrához szükséges számolások elvégzéséért.

IRODALOM

[1] MEDGYESSY, P., „Sűrűségfüggvények és diszkrét eloszlások szuperpozícióinak felbontása", *Magyar Tud. Akad. Mat. Fiz. Oszt. Közl.* **21** (1972) 129—200.

[2] MEDGYESSY, P., „Sűrűségfüggvények és diszkrét eloszlások szuperpozícióinak felbontása. II", *Magyar Tud. Akad. Mat. Fiz. Oszt. Közl.* **21** (1972) 261—382.

[3] MEDGYESSY, P., „Új módszer szimmetrikus sűrűségfüggvények szuperpozícióinak felbontására. I", *Magyar Tud. Akad. Mat. Fiz. Oszt. Közl.* **23** (1974) 33—40.

(Beérkezett: 1976. május 28.)

MEDGYESSY PÁL
MTA MATEMATIKAI KUTATÓ INTÉZET
1053 BUDAPEST V., REÁLTANODA U. 13—15.

Alkalmazott Matematikai Lapok **2** *(1976) 341—349*

A TÖBBDIMENZIÓS NORMÁLIS ELOSZLÁSFÜGGVÉNY MONTE CARLO KISZÁMÍTÁSA AZ ELLIPSZOID MÓDSZER SEGÍTSÉGÉVEL[1]

DEÁK ISTVÁN
Budapest

Egy számítógépes algoritmust közlünk, amely a többdimenziós normális eloszlás eloszlásfüggvénye értékeinek kiszámítására alkalmas. Az algoritmus egy jól ismert *Monte Carlo eljárás*on alapszik és az *Ellipszoid módszer*t használja fel normális eloszlású vektorok generálására. Elkészítettük a megfelelő FORTRAN programot, közöljük a számítógépes tapasztalatokat és a futási időt is.

Az eloszlásfüggvény nagy (1-hez közeli) értékeire gyorsan működik a program, ezek az értékek a sztochasztikus programozásban különösen fontosak. Ha a meghatározandó érték p, akkor az esetek $100p\%$-ában egy normális eloszlású vektor generálása egyetlen (0, 1)-ben egyenletes eloszlású véletlen szám generálására és néhány egyszerű logikai és aritmetikai műveletre egyszerűsödik. Így a számításoknak majdnem $100\%p$-a független lesz a dimenzószámtól.

1. Bevezetés

Feladatunk a következő típusú integrálok meghatározása

$$I_1 = \int_{-\infty}^{h_n} \dots \int_{-\infty}^{h_1} \varphi(\mathbf{x})\, d\mathbf{x} = P\{\xi \in D_1\}, \quad \mathbf{h} \geqq \mathbf{0},$$

(1.1)

$$I_2 = \int_{-h_n}^{h_n} \dots \int_{-h_1}^{h_1} \varphi(\mathbf{x})\, d\mathbf{x} = P\{\xi \in D_2\}, \quad \mathbf{h} \geqq \mathbf{0},$$

ahol ξ egy normális eloszlású véletlen vektor, a D_1 és D_2 tartományok a következők

$$D_1 = \{\mathbf{x} | \mathbf{x} \leqq \mathbf{h},\ \mathbf{h} \geqq \mathbf{0}\},$$

$$D_2 = \{\mathbf{x} | -\mathbf{h} \leqq \mathbf{x} \leqq \mathbf{h},\ \mathbf{h} \geqq \mathbf{0}\},$$

$\varphi(\mathbf{x})$ pedig a $\mathbf{0}$ várható értékű, $\mathbf{R}$ korrelációs mátrixú, n-dimenziós normális eloszlás sűrűségfüggvénye, azaz

$$\varphi(\mathbf{x}) = (2\pi)^{-n/2} |\mathbf{R}|^{-1/2} \exp\left\{-\frac{1}{2}\mathbf{x}'\mathbf{R}^{-1}\mathbf{x}\right\}. \tag{1.2}$$

A cikkben közölt számítógépes algoritmus alkalmazható tetszőleges n dimenziós D tartományra, nemcsak D_1 és D_2 típusúakra. (1.1)-ben az első integrál az eloszlásfüggvény, amelyre gyakran van szükség sztochasztikus programozási modellekben,

[1] Ez a cikk a szerzőnek a IX. International Symposium on Mathematical Programming 1976 augusztus 24-én elhangzott előadását tartalmazza.

nagy I_1 érték esetén [6], [8] [9]. (1.1) második sorában levő integrált statisztikai számításokban és a megbízhatóságelméletben alkalmazzák.

A következő jól ismert *Monte Carlo eljárás* alkalmazható az

$$I = \int_D \dots \int \varphi(\mathbf{x})\, d\mathbf{x} = P\{\xi \in D\}$$

integrál kiszámítására. Generáljuk a $\xi^{(1)}, \dots, \xi^{(N)}$ $\varphi(\mathbf{x})$ sűrűségfüggvényű véletlen vektorokat és számoljuk le, hány fekszik ezek közül a D tartományban. Ha az N vektor közül N_1 darab van D-ben, akkor

$$I = E\left\{\frac{N_1}{N}\right\} \sim \frac{N_1}{N}.$$

Ezt a *Monte Carlo eljárás*t a továbbiakban a valószínűségek kiszámítása „durva" algoritmusának nevezzük.

A következő részben leírjuk az *Ellipszoid módszer*t normális eloszlású vektorok generálására, a 3. részben pedig megmutatjuk, hogyan lehet az *Ellipszoid módszer* sajátosságait kihasználni (1.1) típusú valószínűségek kiszámításában. A 4. részben a számítógépes eredményeket és a futási időket közöljük. Az utolsó részben a közölt algoritmust vizsgáljuk meg, valamint összehasonlítjuk a „durva" módszerrel, mikoris a normális vektorokat mátrix szorzással állítjuk elő.

2. Az Ellipszoid módszer

A módszert [3]-ban közöltük, itt csak röviden ismertetjük. Dekomponáljuk a $\varphi(\mathbf{x})$ sűrűségfüggvényt más sűrűségfüggvények összegeként:

$$\varphi(\mathbf{x}) = p_1 e_1(\mathbf{x}) + \dots + p_k e_k(\mathbf{x}) + p_{k+1} r_1(\mathbf{x}) + p_{k+2} r_2(\mathbf{x}), \tag{2.1}$$

ahol $\sum p_i = 1$, az e_i, $i = 1, \dots, k$, r_1, r_2 függvények n dimenziós eloszlások sűrűségfüggvényei. A (2.1) egyenlőségen alapuló generálási eljárás a következő: válasszuk ki p_i valószínűséggel az e_i vagy r_j sűrűségfüggvények egyikét, a kiválasztott sűrűségfüggvény szerint generáljunk egy n dimenziós vektort és adjuk át azt, mint végső mintát.

Definiáljuk a következő n dimenziós hiperellipszoidokat

$$E_i = \{\mathbf{x} | \varphi(\mathbf{x}) \geqq m_i\}, \quad i = 1, \dots, k,$$

ahol $0 < m_1 < \dots < m_k < \varphi(\mathbf{0})$ a $[0, \varphi(\mathbf{0})]$ intervallum egy felosztása. Definiáljuk most az

$$e_i(\mathbf{x}) = \begin{cases} \dfrac{1}{V_i}, & \mathbf{x} \in E_i, \\ 0, & \mathbf{x} \notin E_i \end{cases}$$

függvényeket, ahol $V_i = \int_{E_i} d\mathbf{x}$ az E_i hiperellipszoid térfogata. A $p_1 + \dots + p_k$ összeg 1-hez tetszőlegesen közelivé tehető egy elég nagy k szám választásával és megfelelő $m_1, \dots, m_k$ felosztással, ahol

$$p_i = V_i(m_i - m_{i-1}), \quad i = 1, \dots, k.$$

Legyenek a μ_i, $i=1, \ldots, k$ multiplikátorok a következők

$$\mu_i = \frac{c_i}{c_1}, \quad i = 1, \ldots, k,$$

ahol a c_i az i-edik hiperellipszoid konstansa, azaz $\mathbf{x}'\mathbf{R}^{-1}\mathbf{x} \leqq c_i^2$, ha $\mathbf{x} \in E_i$.

Ezeknek a multiplikátoroknak a segítségével írhatjuk

$$E_i = \{\mathbf{x} | \mathbf{x} = \mu_i \mathbf{y},\ \mathbf{y} \in E_1\}.$$

Mivel a μ multiplikátor diszkrét eloszlású, $P\{\mu = \mu_i\} = p_i$, $i=1, \ldots, k$, így ezeket egy *Marsaglia táblázat*ban lehet tárolni [4].

A $p_{k+1} r_1(\mathbf{x})$ és a $p_{k+2} r_2(\mathbf{x})$ függvényeket úgy határozzuk meg, hogy a (2.1) egyenlőség igaz legyen; az első a sűrűségfüggvény szélét, a második a végét adja ki ($p_{k+2} r_2(x) = \varphi(x)$, ha $x \notin E_1$).

Az $m_1, \ldots, m_k$ felosztást meg lehet úgy határozni ([3]), hogy a $p_1 + \ldots + p_k \geqq 0{,}98$ reláció teljesül 20 dimenzióban is $k \leqq 700$ esetén. Így az r_1 és r_2 sűrűségfüggvény szerint ritkán generálunk vektort; az algoritmusokat csak az e_i függvényekre írjuk le és megjelöljük, hol kell a megfelelő módosításokat beilleszteni. A $p_{k+1} r_1(\mathbf{x})$ és a $p_{k+2} r_2(\mathbf{x})$ függvényeket elhanyagolva most már le tudjuk írni a (2.1) dekompozíción alapuló generálási eljárást.

E1. Válasszunk ki p_i valószínűséggel egy μ_i multiplikátort a *Marsaglia táblázat*ból. Ez az $e_1, \ldots, e_k$ függvények közül az e_i függvény kiválasztását jelenti.

E2. Generáljunk egy $\mathbf{z}$ vektort egyenletesen az E_1 hiperellipszoidban.

E3. Adjuk át az $\mathbf{x} \leftarrow \mu_i \mathbf{z}$ vektort, mint végső mintát.

3. Valószínűségek kiszámítása az Ellipszoid módszer felhasználásával

Megmutatjuk, hogyan lehet az *Ellipszoid módszer*t kihasználni az

$$I_1 = \int_{-\infty}^{h_n} \ldots \int_{-\infty}^{h_1} \varphi(\mathbf{x})\, d\mathbf{x}, \quad \mathbf{h} \geqq \mathbf{0}$$

típusú valószínűségek kiszámításában.

Két módosítással látjuk el az 1. részben leírt „durva" eljárást. Jelöljük E_c-vel az $\mathbf{x}'\mathbf{R}^{-1}\mathbf{x} \leqq c^2$ egyenletű, a

$$D_1 = \{\mathbf{x} | \mathbf{x} \leqq \mathbf{h},\ \mathbf{h} \geqq \mathbf{0}\}$$

tartományban levő hiperellipszoidok közül a legnagyobbat. Legyen a megfelelő multiplikátor $\mu_c = c/c_1$, ekkor a következő megjegyzéseket tehetjük.

Tegyük fel, hogy az I_1 kiszámítása folyamán egy egyenletes eloszlású vektort kell generálni az E_i hiperellipszoidban (vagyis az E1 lépésben a μ_i multiplikátort választottuk).

Első módosítás. Ha $E_i \subset E_c$, akkor a generálandó vektor E_c-ben fekszik, következésképpen a D_1 tartományban van. Így nem kell az E2 és E3 lépéseket végrehajtanunk, mivel csak az $\mathbf{x} \in D_1$ reláció teljesülésére vagyunk kíváncsiak.

Második módosítás. Ha $E_i \supset E_c$, akkor is várhatjuk, hogy az $\mathbf{x} \in D_1$ reláció némely esetekben ellenőrizhető anélkül, hogy az összes számítást végrehajtanánk; azokban az esetekben, amikor $\mu_i \mathbf{z} \in E_c$, ahol $\mathbf{z}$ egyenletes eloszlású E_1-ben.

Tegyük pontosabbá a második módosítást. Jelöljük $\mathbf{L}$-lel azt az alsó háromszög mátrixot, melyre $LL' = \mathbf{R}$ [10], és legyen S_1 egy n dimenziós hipergömb c_1 sugárral. Ekkor az

$$E_1 = \{\mathbf{x} | \mathbf{x} = \mathbf{L}\mathbf{y},\ \mathbf{y} \in S_1\}$$

összefüggés igaz. Így egy E_1-ben egyenletes eloszlású vektor generálása úgy hajtható végre, hogy (az E2 lépést a következő három lépéssel helyettesítjük):

E 2/a. (Egy véletlen sugárhossz generálása.) Generáljuk u-t egyenletesen (0,1)-ben és legyen $r \leftarrow u^{1/n}$.

E 2/b. Generáljuk $\mathbf{w}$-t egyenletesen az S_1 felszínén.

E 2/c. Az E_1-ben egyenletes eloszlású pont a $\mathbf{z} \leftarrow r\mathbf{L}\mathbf{w}$.

A második módosítást így a $\mu_i r \leqq \mu_c$ feltétel ellenőrzése, közvetlenül az E 2/a lépés végrehajtása után. Ha ez a reláció teljesül, akkor $\mathbf{x} \in E_i \subset D_1$ következik a generálandó $\mathbf{x}$ vektorra.

Az E_c hiperellipszoid multiplikátorának meghatározása

Tegyük fel, hogy az $\mathbf{x}'\mathbf{R}^{-1}\mathbf{x} - 1 = 0$ egyenletű egység hiperellipszoidnak az $x_1 = g_1, \ldots, x_n = g_n$ egyenletekkel meghatározott hipersíkokkal alkotott elsőrendű érintési pontjai az $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}$ pontok, ahol $g_1, \ldots, g_n$ valamilyen konstansok.

A hiperellipszoid normál vektora $\mathbf{R}^{-1}\mathbf{x}$. Mivel a normál vektor az $\mathbf{y}^{(i)}$ érintési pontban párhuzamos $\mathbf{e}^{(i)}$-vel és merőleges $\mathbf{e}^{(j)}$-re, $j \neq i$, ahol $\mathbf{e}^{(i)}$ az i-edik egységvektor, felírhatjuk, hogy

$$\mathbf{e}^{(i)}\mathbf{R}^{-1}\mathbf{y}^{(i)} = 1$$

és

$$\mathbf{e}^{(j)}\mathbf{R}^{-1}\mathbf{y}^{(i)} = 0, \quad j = 1, \ldots, i-1, i+1, \ldots, n.$$

Ezeket egyetlen egyenlőségbe írjuk

$$\mathbf{I}\mathbf{R}^{-1}\mathbf{y}^{(i)} = \mathbf{e}^{(i)},$$

ahol $\mathbf{I}$ az egység mátrix. Mivel ez az utolsó egyenlőség az összes $\mathbf{y}^{(i)}$, $i = 1, \ldots, n$ érintési pontra igaz, így az

$$\mathbf{R}^{-1}(\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}) = \mathbf{I}$$

összefüggést kapjuk.

Tehát az $\mathbf{y}^{(i)}$ érintési pontok az $\mathbf{R} = (\mathbf{r}^{(r)}, \ldots, \mathbf{r}^{(n)})$ korrelációs mátrix oszlopai által meghatározott vektorok, azaz $\mathbf{y}^{(i)} = \mathbf{r}^{(i)}$.

A D_1 tartományon belüli E_c legnagyobb hiperellipszoid meghatározásához tekintsük azt a hipersíkot, amelyet az egység hiperellipszoid $\mu_c > 0$ faktorral történő nagyítása folyamán elsőnek ér el az $x_1 = h_1, \ldots, x_n = h_n$ egyenletekkel meghatározott hipersíkok közül. Mivel az $\mathbf{y}^{(i)}$ érintési pont i-edik koordinátája 1, ezért a keresett hipersík a $\mathbf{h}$ vektor legkisebb komponense által van meghatározva. Így, feltéve, hogy

$$h_m = \min(h_1, \ldots, h_n)$$

az E_c legnagyobb hiperellipszoid konstansa

$$(h_m \mathbf{y}^{(m)})' \mathbf{R}^{-1} (h_m \mathbf{y}^{(m)}) \leqq h_m^2.$$

A keresett multiplikátor ebből az utolsó egyenlőségből

$$\mu_i = \frac{h_m}{c_1}.$$

Most már megfogalmazhatjuk a következő P algoritmust a $P\{\xi \in D_1\}$ valószínűségek kiszámítására. Határozzunk meg egy N számot és legyen $N_1 \leftarrow 0$ a kezdeti érték.

P1. (Határozzunk meg egy E_i hiperellipszoidot.)
Generáljuk u-t egyenletesen (0, 1)-ben és u segítségével válasszunk egy μ_i-t a *Marsaglia táblázatból.*

P2. (Ellenőrizzük, hogy E_i benne van-e E_c-ben.)
Ha $\mu_i < \mu_c$, menjünk P7-re.

P3. (Generáljunk egy véletlen sugárhosszat.)
Generáljuk u-t és legyen $r \leftarrow u^{1/n}$.

P4. (Ellenőrizzük, hogy a generálandó pont E_c-ben lesz-e.)
Ha $\mu_i r < \mu_c$, menjünk P7-re.

P5. Generáljuk **w**-t egyenletesen S_1 felületén és tegyük $\mathbf{x} \leftarrow \mu_i r \mathbf{L}\mathbf{w}$.

P6. (Ellenőrizzük, hogy a pont a D_1 tartományban van-e.)
Ha $\mathbf{x} \in D_1$ menjünk P7-re, egyébként P8-ra.

P7. (Növeljük meg a számlálót.)
Legyen $N_1 \leftarrow N_1 + 1$.

P8. Ismételjük meg az egész eljárást N-szer.

P9. Adjuk át az $I_1 \leftarrow N_1/N$ értéket, mint a D_1 tartomány valószínűségét.

Mivel egy diszkrét eloszlású, véletlen számnak a generálásához a *Marsaglia táblázat* segítségével az egyenletes eloszlású u számnak csak az első kilenc vagy tizenkét bitjére van szükség, az u többi bitje tárolható és a P3 lépésben felhasználható egy új u generálása helyett.

A P1—P4. lépéseket a P algoritmus lineáris részének nevezzük, mivel a szükséges műveletek n-től függetlenek. A lineáris rész valószínűségének nevezzük és p_{lin}-nel jelöljük annak a valószínűségét, hogy a P2 és a P4 lépésben az ellenőrzések eredményeként a P7-re lépünk át közvetlenül. Természetesen

$$p_{\text{lin}} - p_{k+1} \leqq P\{\chi_n^2 \leqq (c_1 \mu_c)^2\} \leqq p_{\text{lin}},$$

ahol p_{k+1} az $r_1(\mathbf{x})$ szél-függvénynek a valószínűsége és χ_n^2 egy n szabadságfokú χ^2 eloszlású változó.

Az $r_1(\mathbf{x})$ és $r_2(\mathbf{x})$ sűrűségfüggvényű véletlen vektorokat, amennyiben generálásuk szükségessé válik, rögtön a P6 lépésben ellenőrizzük.

4. Számítógépes futási idők és eredmények

A P algoritmusnak megfelelő számítógépes program FORTRAN nyelven készült és a KSH ÁSzSz Honeywell 66/20 számítógépén futott.

A program 1800 sor hosszú és 24K memóriát foglal el az összes szükséges szubrutin és makro. Célunk a „durva" módszerhez képest elérhető csökkenés meg-

mutatása volt, így csak a $\mathbf{h}=(h_0, \ldots, h_0)$ esetet vizsgáltuk. A kiszámított integrálok a következők voltak:

$$I_1 = \int_{-\infty}^{h_0} \ldots \int_{-\infty}^{h_0} \varphi(\mathbf{x})\, d\mathbf{x}, \quad h_0 \geqq 0$$

és

$$I_2 = \int_{-h_0}^{h_0} \ldots \int_{-h_0}^{h_0} \varphi(\mathbf{x})\, d\mathbf{x}, \quad h_0 \geqq 0.$$

A fejlécek mutatják, hogy mennyi idő szükséges $N=400$ vektor generálására az *Ellipszoid módszer* segítségével és annak ellenőrzésére, vajon a vizsgált tartományban fekszenek-e vagy sem, azaz a „durva" módszer ideje. Az $N=400$ értéket választottuk, mivel ebben az esetben az I_1 kiszámításának abszolút hibája 0,05-nél kisebb [2]. A táblázatokban megadott idők a P algoritmus idejei, amikoris a két módosítást használtuk a valószínűség kiszámításában.

$n=2$ dimenzió — idő = 1,5 sec

I_1	I_2	p_{lin}	idő (sec)	h_0
0,871	0,730	0,66	0,8	1,5
0,955	0,932	0,85	0,5	2,0
0,987	0,977	0,94	0,3	2,5
0,997	0,993	0,97	0,3	3,0

$n=5$ dimenzió — idő = 3,1 sec

I_1	I_2	p_{lin}	idő (sec)	h_0
0,707	0,477	0,17	2,8	1,5
0,891	0,791	0,44	1,9	2,0
0,969	0,946	0,70	1,2	2,5
0,993	0,990	0,88	0,6	3,0
0,999	0,999	0,95	0,3	3,5

$n=10$ dimenzió — idő = 6 sec

I_1	I_2	p_{lin}	idő (sec)	h_0
0,794	0,640	0,04	5,4	2,0
0,939	0,885	0,19	4,5	2,5
0,986	0,970	0,45	3,3	3,0
0,997	0,995	0,72	1,9	3,5
1,000	1,000	0,89	0,7	4,0

$n=15$ dimenzió — idő = 10,0 sec

I_1	I_2	p_{lin}	idő (sec)	h_0
0,910	0,826	0,02	9,4	2,5
0,979	0,958	0,11	8,5	3,0
0,996	0,995	0,32	6,5	3,5
0,999	0,999	0,60	4,3	4,0
1,000	1,000	0,82	2,1	4,5
1,000	1,000	0,93	1,0	5,0

$n=20$ dimenzió idő = 14,3 sec

I_1	I_2	p_{lin}	idő (sec)	h_0
0,973	0,950	0,01	13,9	3,0
0,995	0,992	0,08	12,4	3,5
0,999	0,999	0,26	10,4	4,0
1,000	1,000	0,54	7,3	4,5
1,000	1,000	0,78	6,6	5,0

Lényegében a lineáris rész valószínűsége határozza meg a számítási idő csökkenését, amint ezt a közölt eredményekből is látni. A táblázatok megmutatják, hogy mikor érdemes az *Ellipszoid módszer*t használni vektorok generálására n dimenziós halmazok valószínűségének meghatározására. (A táblázatok a legjobb eredményeket tartalmazzák abban az értelemben, hogy a $p_{\text{lin}}(h_0)$-nál nem lehet nagyobb a lineáris rész valószínűsége, ha egy $P(\boldsymbol{\xi} \leqq \mathbf{h})$ valószínűséget kell kiszámítani, ahol $\mathbf{h}$-nak legalább egy komponense h_0-val egyenlő, a többi komponense pedig h_0-nál nagyobb.) Például, ha 5 dimenzióban 0,98-nál nagyobb valószínűséget kell kiszámítani, a P algoritmust érdemes használni.

Természetesen a 0,3 sec-os időszintet minden dimenzióban elérhetjük, ha h_0-t elég nagyra választjuk.

5. Megjegyzések és következtetések

A közölt algoritmusról teszünk néhány megjegyzést és összehasonlítjuk más algoritmusokkal.

a) A [2] cikkben közölt eljárásban a normális eloszlású vektorok generálását az általánosan ismert módon végeztük, azaz n darab $N(0, 1)$-es eloszlású független véletlen számot generáltunk és az L alsó háromszög mátrix segítségével transzformáltuk. A [2] cikkben a következőképpen módosítottuk a „durva" módszert. Az L mátrixszal történő szorzást soronként hajtottuk végre olyan sorrendben, hogy az eredményként kapott normális vektornak azokat a komponenseit számítottuk ki és ellenőriztük először, amelyek nagyobb valószínűséggel feküdtek a D_1 tartományon kívül. A módosítás miatt az algoritmus gyorsabban működött kisebb valószínűségek esetén. Minél nagyobb volt a keresett valószínűség, annál több számítási időre volt szükség. 0,9-nél nagyobb valószínűségek esetén a futási idő lényegében ugyanakkora volt, mint a „durva" módszer esetén, sőt, egy kicsivel több is. Így szükségessé vált egy olyan módosítás, amelynek segítségével az algoritmus gyorsan működik nagy valószínűségek esetén. Megemlítünk itt a [2]-ben CDC 3300-as számítógépen kapott néhány futási időt: $n=5$ dimenzió 1 sec, $n=10$ dimenzió 3 sec, $n=20$ dimenzió 9 sec, $n=50$ dimenzió 40 sec ($N=400$ esetén).

b) A leírt P algoritmus nemcsak a megvizsgált, hanem általános n dimenziós D halmazokra is alkalmazható. Tekintsük az $\mathbf{x}'\mathbf{R}^{-1}\mathbf{x} \leqq c^2$ alakú hiperellipszoidokat. Legyen $E_{\min}$ a legkisebb hiperellipszoid, amely tartalmazza D-t és legyen $E_{\max}$ a legnagyobb hiperellipszoid D-n belül. Jelöljük multiplikátorjaikat $\mu_{\min}$ ill. $\mu_{\max}$-szal. Ekkor a P4 lépés helyett a következő két lépés illesztendő be:

P.4/a Ha $\mu_i r < \mu_{\max}$, menjünk P7-re.

P.4/b Ha $\mu_i r > \mu_{\min}$, menjünk $P8$-ra.

P4 helyettesítése a P algoritmusba a fenti két lépéssel magától értetődő.

c) Az S_1 hipergömb felületén egyenletes eloszlású pontok generálását (ld. a P5 lépést) MULLER [7] egy megjegyzése szerint végeztük; a normális eloszlású számokat a *Polár módszer*rel generáltuk a számítógépes programban. Így az összes közölt időt legalább az 1/3-ára lehetne csökkenteni, ha egy gyors módszert használnánk normális eloszlású számok generálására, például AHRENS és DIETER [1] egyik algoritmusát, vagy MARSAGLIA módszerét [5]. További csökkentést lehetne elérni a futási időben a P1—P4 lépések gépi kódban történő megírásával.

A program memóriaigényét azáltal lehetne csökkenteni, hogy a szubrutinokat két overlay-be osztjuk; az egyik az előkészítést végző szubrutinokat tartalmazza, a másik a P algoritmust megvalósítókat.

Ezekkel a módosításokkal a közölt eljárás majdnem ugyanannyi idő alatt futna le kis valószínűségekre, mint a „durva" módszer a háromszög mátrix szorzással.

d) A közölt algoritmus sokkal bonyolultabb, mint a „durva" módszer a háromszög mátrix szorzással. Csak akkor javasolható alkalmazásra, ha nagy számú 1-hez közeli valószínűségeket kell kiszámítani és a 4. részben közölt számok biztosítják a felhasználót, hogy érdemes az algoritmust használni. Az eljárás jó tulajdonságaként a számítógépes program gyorsaságát hangsúlyozzuk 1-hez közeli értékek esetén, megjegyezve, hogy ekkor mind a numerikus, mind pedig a durva módszer munkaigényesebbé válik.

IRODALOM

[1] AHRENS, J. H. and DIETER, U., "Extensions of Forsythe's method for random sampling", *Math. of. Comp.* **27** (1973) 927—937.

[2] DEÁK, I., „A többdimenziós tér halmazai valószínűségének kiszámítása normális eloszlás esetén", *Alk. Mat. L.* **2** (1976) 17—26.

[3] DEÁK, I., "The Ellipsoid method for generating normally distributed random vectors", *Zastosowania Math.* (to appear).

[4] MARSAGLIA, G., "Generating discrete random variables in a computer", *Comm. ACM* **6** (1963) 37—38.

[5] MARSAGLIA, G., MACLAREN, M. D. and BRAY, T. A., "A fast procedure for generating normal random variables", *Comm. ACM* 7 (1964) 4—10.

[6] MAYER, J., "Computational experiences with the reduced gradient method", in: *Coll. Math. Soc. J. Bolyai, 12. Progress in Op. Res. held in Eger (Hungary), 1974* 613—624.

[7] MULLER, M. E., "A note on a method for generating points uniformly on *n*-dimensional spheres", *Comm. ACM* **2** (1959) 19—20.

[8] PRÉKOPA, A. "Application of stochastic programming to engineering design", in: *Proc. of the IX. Int. Conf. on Math. Prog. 1976.*

[9] PRÉKOPA, A., GANZER, S., DEÁK, I. és PATYI, K., „A STABIL sztochasztikus programozási modell kísérleti alkalmazása a magyar villamosenergiaiparra", *Alk. Mat. L.* **1** (1975), 3—22.

[10] SCHEUER, E. M. and STOLLER, D. S., "On the generation of normal random vectors", *Technometrics* **4** (1962) 278—281.

[11] SZÁNTAI, T., „Egy N eljárás a többdimenziós normális eloszlásfüggvény és gradiensének kiszámítására", *Alk. Mat. Lapok* **2** (1976) 27—39.

(Beérkezett: 1977. augusztus 3.)

DEÁK ISTVÁN
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST I., ÚRI U. 49.

THE USE OF THE ELLIPSOID METHOD FOR COMPUTING THE VALUES OF THE MULTIVARIATE NORMAL DISTRIBUTION FUNCTION

I. DEÁK

A computer algorithm is presented for computing the values of the multidimensional normal distribution function. The algorithm is based on a well-known *Monte Carlo technique* and makes use of the *Ellipsoid Method* for generation of random normal vectors. A corresponding FORTRAN program was made, computational results and computer running times are also dealt with.

The program works fast for great values of the distribution function (near to 1) which are especially interesting in stochastic programming models. If the value to be determined is p then in almost $100p\%$ of the cases the generation of one normally distributed vector reduces to the generation of one uniform random number from (0, 1), some minor logical and arithmetical operations. Thus almost $100p\%$ of the work becomes independent of the number of dimensions.

Alkalmazott Matematikai Lapok **2** *(1976) 351—358*

A DUÁLIS MÁTRIXOK MÓDSZERÉNEK EGY OSZTÁLYÁRÓL

ABAFFY JÓZSEF
Budapest

A cikkben megadjuk a duális mátrixok módszerének egy osztályát, amely három, tetszőlegesen választható paramétertől függ. Bebizonyítjuk, hogy egy n változós kvadratikus alak minimumát a paraméterek tetszőleges választása mellett fegfeljebb $n+1$ lépésben megkapjuk. Ez a módszerosztály lehetőséget ad konkrét módszerek kialakítására. Végül megmutatjuk, hogyan adódik speciális esetként két ismert duális mátrix módszer.

1. Bevezetés

Olyan vonalmenti minimalizálást nem használó módszerekkel, amelyek segítségével egy pozitív definit szimmetrikus *Hesse mátrix*szal rendelkező kvadratikus alak minimuma véges lépésben meghatározható, az utóbbi években kezdtek foglalkozni. Ezt az indokolja, hogy nem kvadratikus függvények esetén a vonalmenti minimalizálások nagyon műveletigényesek. E célból fejlesztették ki a duális mátrixok módszerét, amely témakörben jelenleg két cikk ismeretes [3], [4].

Hasonló okok miatt, mint a kvazi-Newton módszerek esetén BROYDEN [1] és HUANG [5] tették, célunk az, hogy megfogalmazzuk a duális mátrix módszerek egy osztályát. Megjegyezzük még, hogy a kidolgozandó módszerosztály nem szimmetrikus típusú eljárásokat is tartalmaz, de ezzel kapcsolatban hivatkozunk BROYDEN, DENNIS és MORÉ [2] cikkére, amelyben a szerzők megmutatják, hogy „direct prediction" módszerek esetén a nem szimmetrikus *Pearson módszer* lokálisan konvergens, ugyanakkor létezik olyan egy rangú szimmetrikus módszer, amely nem az. Minthogy a kidolgozandó módszerosztály esetén vonalmenti minimalizálást nem használunk, így úgy gondoljuk a nem szimmetrikus eljárások is érdekesek lehetnek.

2. A duális mátrix módszerek egy osztályának leírása

A duális mátrixok módszerének megértése végett induljunk ki HUANG [4] cikkéből.

Tekintsük a következő kvadratikus függvényt

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c, \quad \mathbf{x}, \mathbf{b} \in R^n, c \in R^1, \tag{2.1}$$

ahol $\mathbf{A}$ $n \times n$-es pozitív szemidefinit szimmetrikus mátrix, és $r(\mathbf{A}) = m \leqq n$. Az $f(\mathbf{x})$ gradinese az $\mathbf{x} \in R^n$ helyen

$$\mathbf{g}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}. \tag{2.2}$$

Legyen $\mathbf{y}_i = \mathbf{g}_{i+1} - \mathbf{g}_i$ és $\mathbf{s}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$, ahol $\mathbf{g}_i$ $f(\mathbf{x})$ gradiense az $\mathbf{x}_i \in R^n$ $(i = 1, 2, \ldots, n)$ helyen, az $\mathbf{x}_i$, $i = 1, 2, \ldots, n$ pontsorozatot később definiáljuk.

(2.2)-ből következik, hogy

$$\mathbf{y}_i = \mathbf{A}\mathbf{s}_i, \quad i = 1, 2, \ldots, n. \tag{2.3}$$

Minthogy $\mathbf{A}$ rangja $m \leqq n$, nyilván maximálisan m lineárisan független $\mathbf{y}_i$ létezik.

Legyen az $\mathbf{x}_i$, $i = 1, 2, \ldots, n$ sorozat meghatározva úgy, hogy az $\mathbf{s}_i$, $i = 1, 2, \ldots, n$ irányokra teljesüljön

$$\mathbf{s}_i^T \mathbf{A} \mathbf{s}_j = 0, \quad 1 \leqq i \leqq j-1, \tag{2.4}$$

azaz az $\mathbf{s}_i$ irányok $\mathbf{A}$-konjugáltak legyenek.

Tegyük fel, hogy az l-edik lépésben kapott $\mathbf{y}_l$ lineáris kombinációja az előző $\mathbf{y}_i$ irányoknak. Ekkor a következő lépésben megkaphatjuk $f(\mathbf{x})$ minimumát. Minthogy

$$\mathbf{g}_l = \sum_{i=1}^{l-1} c_i \mathbf{y}_i, \quad c_i \in R^1, \quad i = 1, 2, \ldots, l-1, \tag{2.5}$$

a (2.3) egyenlőség miatt

$$\mathbf{g}_l = \sum_{i=1}^{l-1} c_i \mathbf{A}\mathbf{s}_i.$$

Megszorozva ezt balról $\mathbf{s}_i^T$-vel, és felhasználva (2.4)-et kapjuk, hogy

$$\mathbf{g}_l = \sum_{i=1}^{l-1} \frac{\mathbf{s}_i^T \mathbf{g}_l}{\mathbf{s}_i^T \mathbf{y}_i} \mathbf{A}\mathbf{s}_i.$$

Legyen

$$\mathbf{s}_l = -\sum_{i=1}^{l-1} \frac{\mathbf{s}_i \mathbf{s}_i^T}{\mathbf{s}_i^T \mathbf{y}_i} \mathbf{g}_l. \tag{2.6}$$

Akkor $\mathbf{g}_{l+1}$-re a következőt kapjuk

$$\mathbf{g}_{l+1} = \mathbf{g}_l + \mathbf{A}\mathbf{s}_l = \sum_{i=1}^{l-1} \frac{\mathbf{s}_i^T \mathbf{g}_l}{\mathbf{s}_i^T \mathbf{y}_i} \mathbf{A}\mathbf{s}_i - \mathbf{A} \sum_{i=1}^{l-1} \frac{\mathbf{s}_i \mathbf{s}_i^T}{\mathbf{s}_i^T \mathbf{y}_i} \mathbf{g}_l = \mathbf{0}.$$

Azaz valóban az $\mathbf{x}_{l+1} = \mathbf{x}_l + \mathbf{s}_l$ hely $f(\mathbf{x})$ minimumhelye, mert $\mathbf{g}_{l+1} \equiv \mathbf{0}$.

A fentiekből következik, hogy ha $m = n$, akkor legfeljeb $n+1$ lépésben megkapjuk a minimumot. A (2.6) kifejezésből világos, hogy $\mathbf{s}_l$ meghatározásához szükségünk van egy mátrix sorozatra, amely a következőképpen van definiálva

$$\mathbf{B}_{i+1} = \mathbf{B}_i + \frac{\mathbf{s}_i \mathbf{s}_i^T}{\mathbf{s}_i^T \mathbf{y}_i}, \quad i = 1, 2, \ldots, n, \quad \text{és} \quad \mathbf{B}_1 \tag{2.7}$$

zérus mátrix.

Azonnal látható, hogy a $\mathbf{B}_i$ mátrixsorozat a következő tulajdonsággal rendelkezik

$$\mathbf{B}_{i+1} \mathbf{y}_j = \mathbf{s}_j, \quad 1 \leqq j \leqq i, \quad i = 1, 2, \ldots, n. \tag{2.8}$$

A (2.8) tulajdonság alapján az előzőnél általánosabb mátrix sorozatot határozunk meg.

Keressük most $\mathbf{B}_{i+1}$-et a $\mathbf{B}_{i+1}=\mathbf{B}_i+\mathbf{C}_i$ alakban. Innen

$$\mathbf{B}_{i+1}\mathbf{y}_j = \mathbf{B}_i\mathbf{y}_j+\mathbf{C}_i\mathbf{y}_j = \mathbf{s}_j, \quad j = 1, 2, \ldots, i. \tag{2.9}$$

Megjegyezzük a $\mathbf{C}_i$ mátrix következő tulajdonságát. Jelölje $\mathbf{Y}_j$ $(1\leqq j<i-1)$ az $\mathbf{y}_j$ vektorokból képezett mátrixot. Akkor $\mathbf{C}_i\mathbf{Y}_j=\mathbf{0}$ $(j=1, 2, \ldots, i-1)$ a (2.8) feltétel és (2.9) miatt. Legyenek a $\mathbf{z}_j$, $\mathbf{q}_j$ irányok olyanok, hogy

$$\mathbf{z}_j^T\mathbf{y}_j = 1, \quad \mathbf{q}_j^T\mathbf{y}_j = 1, \quad j = 1, 2, \ldots, i, \tag{2.10}$$

és definiáljuk a $\mathbf{C}_i$ mátrixot a következőképpen

$$\mathbf{C}_i = \mathbf{s}_j\mathbf{q}_j^T-\mathbf{B}_i\mathbf{y}_j\mathbf{z}_j^T, \quad j = 1, 2, \ldots, i.$$

Az előbb említett tulajdonsága a $\mathbf{C}_i$ mátrixnak természetesen ilyen választás mellett is fennáll.

Megköveteljük, hogy a $\mathbf{z}_j$, $\mathbf{q}_j$ irányok olyanok legyenek, amelyekre (2.10) teljesül.

Legyen például

$$\mathbf{q}_j^T = \frac{1-\beta_j\mathbf{y}_j^T\mathbf{B}_j\mathbf{y}_j}{\mathbf{s}_j^T\mathbf{y}_j}\mathbf{s}_j^T+\beta_j\mathbf{y}_j^T\mathbf{B}_j, \quad \beta_j\in R^1, \quad j = 1, 2, \ldots \tag{2.11}$$

és

$$\mathbf{z}_j^T = \frac{\delta_j\mathbf{y}_j^T\mathbf{B}_j\mathbf{y}_i+1}{\mathbf{s}_j^T\mathbf{y}_j}\mathbf{s}_j^T-\delta_j\mathbf{y}_j^T\mathbf{B}_j, \quad \delta_j\in R^1, \quad j = 1, 2, \ldots. \tag{2.12}$$

Felhasználva (2.11)-et és (2.12)-t, $\mathbf{B}_{j+1}$-re a következőt kapjuk:

(2.13)

$$\mathbf{B}_{j+1} = \mathbf{B}_j+\frac{1-\beta_j\mathbf{y}_j^T\mathbf{B}_j\mathbf{y}_j}{\mathbf{s}_j^T\mathbf{y}_j}\mathbf{s}_j\mathbf{s}_j^T+\beta_j\mathbf{s}_j\mathbf{y}_j^T\mathbf{B}_j-\frac{\delta_j\mathbf{y}_j^T\mathbf{B}_j\mathbf{y}_j+1}{\mathbf{s}_j^T\mathbf{y}_j}\mathbf{B}_j\mathbf{y}_j\mathbf{s}_j^T+\delta_j\mathbf{B}_j\mathbf{y}_j\mathbf{y}_j^T\mathbf{B}_j.$$

2.1. TÉTEL. A $\mathbf{B}_j$ mátrixsorozat (2.13) szerinti megválasztása tetszőleges β_j, δ_j konstansok mellett kielégíti a (2.8) feltételt.

Bizonyítás. Tetszőleges $j\geqq 1$ indexre a (2.10) relációból, a (2.9) kifejezésből és a $\mathbf{C}_i$ definíciójából azonnal adódik, hogy

$$\mathbf{B}_{j+1}\mathbf{y}_j = \mathbf{s}_j. \tag{2.14}$$

Tegyük fel most, hogy

$$\mathbf{B}_j\mathbf{y}_i = \mathbf{s}_i, \quad 1 \leqq i < j, \tag{2.15}$$

és lássuk be (2.15) helyességét $j=j+1$ esetén. Kihasználva $\mathbf{C}_i$ definícióját és az előbbiekben leírt tulajdonságát, a $\mathbf{q}_j$, $\mathbf{z}_j$ vektorok (2.10) feltételnek megfelelő megválasztását, valamint a (2.14) tulajdonságot, kapjuk, hogy

$$\mathbf{B}_{j+1}\mathbf{y}_i = \mathbf{B}_j\mathbf{y}_i+\mathbf{C}_j\mathbf{y}_i = \mathbf{B}_j\mathbf{y}_i = \mathbf{s}_i, \quad 1 \leqq i \leqq j, \tag{2.16}$$

amivel a 2.1 tétel állítását beláttuk.

Megjegyezzük, hogy a (2.10) feltételek választása helyett a gyengébb

$$\mathbf{z}_j^T\mathbf{y}_j \neq 0 \quad \text{és} \quad \mathbf{q}_j^T\mathbf{y}_j \neq 0$$

megkötések a (2.10) feltételekre vezetnek vissza, hiszen a $\mathbf{C}_i$ mátrixok (2.9)-ből következő tulajdonsága és $\mathbf{C}_i$ definíciója miatt a két konstansnak meg kell egyeznie.

Felmerül továbbá, hogy a (2.11) és (2.12) kifejezésekben a $\mathbf{q}_j^T$-t illetve $\mathbf{z}_j^T$-t lehetne-e az $\mathbf{s}_j$ és $\mathbf{y}_j^T\mathbf{B}_j$ vektorok tetszőleges lineáris kombinációjaként megválasztani.

Legyen például

$$\mathbf{q}_j^T = \varepsilon_j \mathbf{s}_j^T + \vartheta_j \mathbf{y}_j^T \mathbf{B}_j.$$

A (2.10) megfelelő feltétele miatt a

$$\mathbf{q}_j^T \mathbf{y}_j = \varepsilon_j \mathbf{s}_j^T \mathbf{y}_j + \vartheta_j \mathbf{y}_j^T \mathbf{B}_j \mathbf{y}_j = 1$$

egyenlőségnek kell teljesülnie minden j-re, emiatt az

$$\varepsilon_j = \frac{1 - \vartheta_j \mathbf{y}_j^T \mathbf{B}_j \mathbf{y}_j}{\mathbf{s}_j^T \mathbf{y}_j}$$

összefüggésnek kell fennállnia ε_j és ϑ_j között, ami a (2.11) választást jelenti.

Az $\mathbf{x}_1, \mathbf{x}_2, \ldots$ pontsorozatot úgy kell meghatároznunk, hogy a belőlük meghatározott $\mathbf{s}_1, \mathbf{s}_2, \ldots$ irányok kielégítsék a (2.4) feltételt. Az $\{\mathbf{x}_i\}$ pontsorozatot a következő formában keressük.

Legyen $\mathbf{H}_1$ pozitív definit mátrix, $\mathbf{x}_1 \in R^n$ tetszőleges vektor,

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{s}_i, \quad \mathbf{s}_i = -\alpha_i \mathbf{H}_i^T \mathbf{g}_i, \quad \alpha_i \neq 0, \quad \alpha_i \in R^1, \quad i = 1, 2, \ldots.$$

Az α_i-re nem tesszük fel, hogy minimalizáló az $\mathbf{s}_i$ irányban!

A (2.4) feltételt a következőképpen alakíthatjuk át:

$$\mathbf{s}_i^T \mathbf{A} \mathbf{s}_j = -\alpha_i \mathbf{g}_i^T \mathbf{H}_i \mathbf{y}_j = 0. \tag{2.17}$$

A (2.17) kifejezés alapján a $\mathbf{H}_i$ mátrixot úgy kell meghatároznunk, hogy

$$\mathbf{H}_i \mathbf{y}_j = \mathbf{0}, \quad 1 \leqq j \leqq i-1, \quad i = 2, 3, \ldots \tag{2.18}$$

fennálljon, így ugyanis a $\mathbf{g}_i \neq \mathbf{0}$ irányra mindegyik ortogonális lesz. A $\mathbf{H}_i$ mátrixot szintén a $\mathbf{H}_i = \mathbf{H}_{i-1} + \mathbf{D}_{i-1}$ rekurzív alakban keressük.

A (2.18) feltétel miatt

$$\mathbf{H}_{i-1}\mathbf{y}_j + \mathbf{D}_{i-1}\mathbf{y}_j = \mathbf{0}, \quad 0 \leqq j \leqq i-2, \quad i = 2, 3, \ldots.$$

Legyen a $\mathbf{w}_j$ irány olyan, hogy a $\mathbf{w}_j^T \mathbf{y}_j = 1$ egyenlőség fennálljon minden j-re, és definiáljuk $\mathbf{D}_{i-1}$-et a következőképpen

$$\mathbf{D}_{i-1} = -\mathbf{H}_{i-1}\mathbf{y}_j \mathbf{w}_j^T.$$

Legyen $\mathbf{w}_j^T$ a következő

$$\mathbf{w}_j^T = -\gamma_j \mathbf{s}_j^T + \frac{\gamma_j \mathbf{s}_j^T \mathbf{y}_j + 1}{\mathbf{y}_j^T \mathbf{H}_j \mathbf{y}_j} \mathbf{y}_j^T \mathbf{H}_j. \tag{2.19}$$

A (2.19) alapján kapjuk, hogy

$$\mathbf{H}_i = \mathbf{H}_{i-1} + \gamma_{i-1} \mathbf{H}_{i-1} \mathbf{y}_{i-1} \mathbf{s}_{i-1}^T - \frac{\gamma_{i-1} \mathbf{s}_{i-1}^T \mathbf{y}_{i-1} + 1}{\mathbf{y}_{i-1}^T \mathbf{H}_{i-1} \mathbf{y}_{i-1}} \mathbf{H}_{i-1} \mathbf{y}_{i-1} \mathbf{y}_{i-1}^T \mathbf{H}_{i-1}. \tag{2.20}$$

2.2. TÉTEL. Ha $\mathbf{H}_1$ pozitív definit mátrix, akkor a (2.20) által meghatározott mátrixsorozat teljesíti a (2.18) feltételt.

Bizonyítás. Először belátjuk, hogy

$$\mathbf{H}_2\mathbf{y}_1 = \mathbf{0}. \tag{2.21}$$

A definíció szerint

$$\mathbf{H}_2\mathbf{y}_1 = \mathbf{H}_1\mathbf{y}_1 + \gamma_1\mathbf{H}_1\mathbf{y}_1\mathbf{s}_i^T\mathbf{y}_1 - \frac{\gamma_1\mathbf{s}_i^T\mathbf{y}_1+1}{\mathbf{y}_1^T\mathbf{H}_1\mathbf{y}_1}\mathbf{H}_1\mathbf{y}_1\mathbf{y}_1^T\mathbf{H}_1\mathbf{y}_1 = \mathbf{0}. \tag{2.22}$$

Feltesszük, hogy fennáll

$$\mathbf{H}_i\mathbf{y}_k = \mathbf{0}, \quad 1 \leqq k \leqq i-1, \tag{2.23}$$

és belátjuk, hogy (2.23) igaz $i=i+1$-re is. Ez két részletben történik. Alkalmazva (2.22)-t úgy, hogy a 2-es index helyébe $i+1$-et, az 1-es index helyébe i-t írunk, következik, hogy

$$\mathbf{H}_{i+1}\mathbf{y}_i = \mathbf{0}. \tag{2.24}$$

Elegendő tehát belátnunk, hogy

$$\mathbf{H}_{i+1}\mathbf{y}_k = \mathbf{0}, \quad 1 \leqq k < i.$$

Felhasználva (2.20)-at kapjuk, hogy

$$\mathbf{H}_{i+1}\mathbf{y}_k = \mathbf{H}_i\mathbf{y}_k + \gamma_i\mathbf{H}_i\mathbf{y}_i\mathbf{s}_i^T\mathbf{y}_k - \frac{\gamma_i\mathbf{s}_i^T\mathbf{y}_i+1}{\mathbf{y}_i^T\mathbf{H}_i\mathbf{y}_i}\mathbf{H}_i\mathbf{y}_i\mathbf{y}_i^T\mathbf{H}_i\mathbf{y}_k, \quad 1 \leqq k < i.$$

A (2.23) indukciós feltevés miatt $\mathbf{H}_i\mathbf{y}_k=\mathbf{0}$, másrészt

$$\mathbf{s}_i^T\mathbf{y}_k = -\alpha_i\mathbf{g}_i^T\mathbf{H}_i\mathbf{y}_k = 0$$

szintén fennáll, ezért

$$\mathbf{H}_{i+1}\mathbf{y}_k = 0, \quad 1 \leqq k < i. \tag{2.25}$$

A (2.24) és a (2.25) egyenlőségek bizonyítják a tétel állítását.

Az eddigiek alapján megfogalmazhatjuk a duális mátrixok egy általános osztályát, amely az $f(\mathbf{x})$ kvadratikus függvény minimumát legfeljebb $m+1$ lépésben meghatározza, és ez az osztály olyan, hogy az $\mathbf{s}_i$ irány mentén nem szükséges vonalmenti minimalizálást végeznünk, csupán egy nem zérus lépést kell tennünk.

Legyen $\mathbf{H}_1$ pozitív definit mátrix, $\mathbf{x}_1 \in R^n$ olyan, hogy $\mathbf{g}(\mathbf{x}_1) \neq \mathbf{0}$, és B_1 tetszőleges mátrix. Ekkor az osztályt a következő algoritmus definiálja:

$$\mathbf{s}_i = -\alpha_i\mathbf{H}_i^T\mathbf{g}_i \tag{2.26}$$

$\alpha_i \neq 0$ tetszőleges, illetve nem kvadratikus függvénye olyan, hogy $f(\mathbf{x}_{i+1}) < f(\mathbf{x}_i)$ teljesüljön;

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{s}_i, \quad i = 1, 2, \ldots, \tag{2.27}$$

$$\mathbf{H}_{i+1} = \mathbf{H}_i + \gamma_i\mathbf{H}_i\mathbf{y}_i\mathbf{s}_i^T - \frac{\gamma_i\mathbf{s}_i^T\mathbf{y}_i+1}{\mathbf{y}_i^T\mathbf{H}_i\mathbf{y}_i}\mathbf{H}_i\mathbf{y}_i\mathbf{y}_i^T\mathbf{H}_i, \quad i = 1, 2, \ldots, \tag{2.28}$$

$$\mathbf{B}_{i+1} = \mathbf{B}_i + \frac{1-\beta_i\mathbf{y}_i^T\mathbf{B}_i\mathbf{y}_i}{\mathbf{s}_i^T\mathbf{y}_i}\mathbf{s}_i\mathbf{s}_i^T + \beta_i\mathbf{s}_i\mathbf{y}_i^T\mathbf{B}_i - \frac{\delta_i\mathbf{y}_i^T\mathbf{B}_i\mathbf{y}_i+1}{\mathbf{s}_i^T\mathbf{y}_i}\mathbf{B}_i\mathbf{y}_i\mathbf{s}_i^T + \delta_i\mathbf{B}_i\mathbf{y}_i\mathbf{y}_i^T\mathbf{B}_i,$$

$$i = 1, 2, \ldots, \tag{2.29}$$

és ha egy $\mathbf{g}_l$ irány az $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{l-1}$ lineáris kombinációja, akkor

$$\mathbf{s}_l = -\mathbf{B}_l \mathbf{g}_l. \tag{2.30}$$

Minthogy az algoritmus szabad paraméterekkel rendelkezik $(\gamma_i, \beta_i, \delta_i)$, lehetőség van a paraméterek lépésenkénti alkalmas megválasztására.

3. Az általános séma szimmetrikus esete

Ebben a pontban leírjuk az általános sémának azt az esetét, amikor mind a $\mathbf{H}_i$, mind a $\mathbf{B}_i$ mátrixsorozat szimmetrikus.

Ha $\gamma_i = 0$, $i = 1, 2, \ldots$, akkor (2.88) helyett kapjuk

$$\mathbf{H}_{i+1} = \mathbf{H}_i - \frac{\mathbf{H}_i \mathbf{y}_i \mathbf{y}_i^T \mathbf{H}_i}{\mathbf{y}_i^T \mathbf{H}_i \mathbf{y}_i}, \quad i = 1, 2, \ldots. \tag{3.1}$$

Amennyiben pedig $\beta_i = -\dfrac{\delta_i \mathbf{y}_i^T \mathbf{B}_i \mathbf{y}_i + 1}{\mathbf{s}_i^T \mathbf{y}_i}$, akkor

$$\mathbf{B}_{i+1} = \mathbf{B}_i + \frac{1-\beta_i \mathbf{y}_i^T \mathbf{B}_i \mathbf{y}_i}{\mathbf{s}_i^T \mathbf{y}_i} \mathbf{s}_i \mathbf{s}_i^T + \beta_i (\mathbf{s}_i \mathbf{y}_i^T \mathbf{B}_i + \mathbf{B}_i \mathbf{y}_i \mathbf{s}_i^T) - \frac{\beta_i \mathbf{s}_i^T \mathbf{y}_i + 1}{\mathbf{y}_i^T \mathbf{B}_i \mathbf{y}_i} \mathbf{B}_i \mathbf{y}_i \mathbf{y}_i^T \mathbf{B}_i, \tag{3.2}$$
$$i = 1, 2, \ldots.$$

Tehát szimmetrikus esetben a $\mathbf{H}_i$ mátrixsorozat egyértelműen meghatározott, míg a $\mathbf{B}_i$ mátrixsorozatban a β_i paramétert tetszőlegesen választhatjuk.

4. Az ismert módszerek meghatározására az általános sémából

Az eddigiekben publikált duál mátrix módszerek szimmetrikusak. Mindegyik módszer a (3.1) kifejezést használja a $\mathbf{H}_i$ mátrixsorozat meghatározására. Amennyiben (3.2)-ben $\beta_i = 0$, $i = 1, 2, \ldots$ akkor a HUANG cikkben közölt II. algoritmust kapjuk, amelyben

$$\mathbf{B}_{i+1} = \mathbf{B}_i + \frac{\mathbf{s}_i \mathbf{s}_i^T}{\mathbf{s}_i^T \mathbf{y}_i} - \frac{\mathbf{B}_i \mathbf{y}_i \mathbf{y}_i^T \mathbf{B}_i}{\mathbf{y}_i^T \mathbf{B}_i \mathbf{y}_i}, \quad i = 1, 2, \ldots.$$

Amennyiben $\beta_i = -\dfrac{1}{(\mathbf{s}_i - \mathbf{B}_i \mathbf{y}_i)^T \mathbf{y}_i}$, akkor pedig a HUANG cikkben közölt III. algoritmust kapjuk, amelyben

$$\mathbf{B}_{i+1} = \mathbf{B}_i + \frac{(\mathbf{s}_i - \mathbf{B}_i \mathbf{y}_i)(\mathbf{s}_i - \mathbf{B}_i \mathbf{y}_i)^T}{(\mathbf{s}_i - \mathbf{B}_i \mathbf{y}_i)^T \mathbf{y}_i}, \quad i = 1, 2, \ldots.$$

A további ismert algoritmusok az eddigiekhez teljesen hasonló módon adódnak.

5. Feltétel a (2.30) lépés választására, és folytatás nem kvadratikus függvények esetén

Minthogy számítógépen a gradiens irány lineáris függősége az $\mathbf{y}_1, \mathbf{y}_2, \ldots$ irányoktól csak egy bizonyos pontosságon belül dönthető el, így az összefüggőséget előre adott $\varepsilon > 0$ esetre kell eldöntenünk. HUANG cikkében erre a következő feltételt kapta:

Ha egy l indexre

$$\left| \frac{\mathbf{g}_l^T \mathbf{s}_l}{\mathbf{g}_l^T \mathbf{g}_l} \right| < \varepsilon$$

akkor a (2.30) lépéssel kell folytatnunk az algoritmust. Minthogy nem kvadratikus függvény esetén az eljárás ezzel a lépéssel valószínűleg nem fejeződik be, meg kell határoznunk a folytató $\mathbf{H}_l$, $\mathbf{B}_l$ mátrixokat is. Minthogy a $\mathbf{H}_1$, $\mathbf{B}_1$ mátrixok pozitiv definitek, ezért, amennyiben a $\mathbf{B}_l$ mátrixsorozat szimmetrikus és $\delta_i \leqq 0$ élhetünk a

$$\mathbf{H}_{l+1} = \mathbf{B}_l,$$

$$\mathbf{B}_{l+1} = \mathbf{B}_l$$

választással.

A (3.2) kifejezés ugyanis éppen a *Broyden osztályt* határozza meg, amelynek pozitív definitségét BROYDEN bebizonyította. Minthogy a *Broyden osztály* a $\beta_i = -\vartheta_i$, $i = 1, 2, \ldots$ választással adódik, így pozitív definit és szimmetrikus mátrixsorozatot kapunk, ha

$$\beta_i \leqq 0, \quad i = 1, 2, \ldots .$$

Minden más esetben a

$$\mathbf{H}_l = \mathbf{H}_1$$

$$\mathbf{B}_l = \mathbf{B}_1$$

választással élhetünk.

IRODALOM

[1] BROYDEN, C. G., "Quasi-Newton Methods and their Application to Function Minimisation" *Math. of. Comp.* **21** (1967) 368—381.

[2] BROYDEN, C. G., Dennis J. E., Moré J. J. "On the Local and Superlinear Convergence of Quasi-Newton Methods" *J. Inst. Maths. Applics.* **12** (1973) 223—245.

[3] BASS, R., "A Rank Two Algorithm for Unconstrained Minimization" *Math. of Comp.* **26** (1972) 129—143.

[4] HUANG, H. Y., "Unified Approach to Quadratically Convergent Algorithm for Function Minimization" *JOTA* **5** (1970) 405—423.

[5] HUANG, H. Y., "Method of Dual Matrices for Function Minimization" *JOTA* **13** (1974) 519—537.

(Beérkezett: 1975. július 10.)

(Újra beérkezett: 1976. június 23.)

ABAFFY JÓZSEF
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST I., ÚRI U. 49.

ON A CLASS OF THE METHODS OF DUAL MATRICES

J. ABAFFY

A class of the method of dual matrices, depending on three parameters which can be chosen arbitrarily, is discussed in the paper. It is proven that the minimum of a quadratic form with n variables is obtained in at most $n+1$ steps under arbitrary choice of parameters. This method class enables the development of concrete methods. Finally it is shown how two known dual-matrix methods result as special cases.

Alkalmazott Matematikai Lapok **2** *(1976) 359—362*

MÁTRIXINVERTÁLÓ MÓDSZEREKRŐL

GERGELY JÓZSEF
Budapest

A dolgozat a *Gauss—Jordan elimináció* és a *rendszámnöveléses mátrixinverzió* megegyezését bizonyítja, majd ennek néhány következményét vizsgálja.

1. Bevezetés

Mátrixok invertálására nagyon sok módszer ismeretes. Ezek közül a leggyakrabban használtak a *Gauss-elimináció,* a *Gauss—Jordan-elimináció,* a *trianguláris felbontással való invertálás* és *rendszámnöveléssel történő invertálás.* Több irodalmi forrásban megtalálható, hogy a felsorolt módszerek közül az első három ekvivalens, (lásd pl. [2], [3]). Az alábbiakban bizonyítjuk, hogy a *rendszámnöveléses módszer* is ekvivalens a többivel, pontosabban a *Gauss—Jordan-eliminációs módszer* és a *rendszámnöveléses invertáló módszer* azonosságát bizonyítjuk be.

Legyen az invertálandó mátrixunk $\mathbf{A}=\{a_{ij}\}$, $i, j=1, \ldots, n$ nemszinguláris, és jelölje $\mathbf{A}_k$ az $\mathbf{A}$ mátrix k-adrendű bal felső szeletét, tehát az a_{ij}, $i, j=1, 2, \ldots, k$ elemekből álló minormátrixot ($k=1, 2, \ldots, n$). Tegyük fel, hogy az összes A_k nem szinguláris.

A *rendszámnöveléses módszer* a következő (lásd [5]). Keressük az

$$\mathbf{A}_k = \begin{pmatrix} \mathbf{A}_{k-1} & \mathbf{u}_k \\ \mathbf{v}_k^T & \alpha_k \end{pmatrix}$$

mátrix inverzét

$$\mathbf{A}_{k-1}^{-1} = \begin{pmatrix} \mathbf{P}_{k-1} & \mathbf{r}_k \\ \mathbf{q}_k^T & \beta_k \end{pmatrix}$$

alakban, ahol $\mathbf{A}_1=a_{11}$, $(\mathbf{A}_1^{-1}=1/a_{11})$. Az inverzet az alábbi képletekkel számolhatjuk:

$$\beta_k = \frac{1}{\alpha_k - \mathbf{v}_k^T \mathbf{A}_{k-1}^{-1} \mathbf{u}_k}, \quad \mathbf{r}_k = -\beta_k \mathbf{A}_{k-1}^{-1} \mathbf{u}_k, \quad \mathbf{q}_k^T = -\beta_k \mathbf{v}_k^T \mathbf{A}_{k-1}^{-1},$$

$$\mathbf{P}_{k-1} = \mathbf{A}_{k-1}^{-1} + \beta_k \mathbf{A}_{k-1}^{-1} \mathbf{u}_k \mathbf{v}_k^T \mathbf{A}_{k-1}. \tag{1.1}$$

Ha az (1.1) képleteket végigszámoljuk $k=2, \ldots, n$-re, akkor $\mathbf{A}_n^{-1}=\mathbf{A}^{-1}$ a keresett inverz.

A *Gauss—Jordan-elimináció* i-edik lépését a következő képletek adják meg:

$$
\begin{aligned}
a_{ii} &:= 1/a_{ii} \\
a_{ij} &:= -a_{ij}a_{ii}, \quad j = 1, \ldots, n, \quad j \neq i; \\
a_{kl} &:= a_{kl} + a_{ki}a_{il}, \quad k, l = 1, \ldots, n, \quad k, l \neq i; \\
a_{ji} &:= a_{ji}a_{ii}, \quad j = 1, \ldots, n, \quad j \neq i.
\end{aligned} \tag{1.2}
$$

A := jel azt jelenti, hogy számítsuk ki a jeltől jobbra álló kifejezés értékét is írjuk a jel baloldalán álló mennyiség helyére. Ha az (1.2) összefüggéseket végigszámoljuk $i=1, \ldots, n$-re, a kiindulási $\mathbf{A}=\{a_{ij}\}$ mátrix helyén az $\mathbf{A}^{-1}$ inverzet kapjuk.

2. A tétel bizonyítása

2.1. TÉTEL. Az **A** mátrix invertálása során az (1.1)-gyel számolt *rendszámnöveléses* és az (1.2)-vel számolt *Gauss—Jordan-eliminációs* módszer identikusan azonos műveleteket végez.

Bizonyítás. Nyilvánvaló, hogy az (1.1) eljárás a k-adik ($2 \leqq k \leqq n$) lépése után az **A** mátrix k-adrendű bal felső szeletének inverzét az $\mathbf{A}_k^{-1}$ mátrixot adja. Pontosan ugyanezt az inverzet kapjuk az (1.2) eljárás k-adik lépése után is az **A** mátrix k-adrendű bal felső szeletében. Vagyis a k-adik ($2 \leqq k \leqq n$) lépés után az (1.1) és az (1.2) eljárás ugyanazt a mátrixot eredményezi az **A** mátrix bal felső szeletében. Bizonyítani akarjuk, hogy amikor $k=m-1 \leqq n-1$-edik állapotból a $k+1=m$-edik lépést végezzük az (1.1) és az (1.2) eljárással számolva ugyanazokat a műveleteket végezzük.

(i) Legyen $m-1<n$ és az $m-1$-edik lépésről térjünk át az m-edik lépésre. Ismeretes (lásd pl. [4]), hogy (1.2)-ben

$$a_{mm} := 1/a_{mm} = \frac{\det(\mathbf{A}_{m-1})}{\det(\mathbf{A}_m)}. \tag{2.1}$$

Számítsuk ki β_m-et (1.1) szerint. Ez az $\mathbf{A}_m^{-1}$ mátrix m, m indexű eleme, $(\mathbf{A}_m^{-1})_{mm}$, tehát ismert mátrixelméleti összefüggések szerint

$$\beta_m = (\mathbf{A}_m^{-1})_{mm} = \frac{1}{\det(\mathbf{A}_m)} (\operatorname{adj} \mathbf{A}_m)_{mm} = \frac{\det(\mathbf{A}_{m-1})}{\det(\mathbf{A}_m)}. \tag{2.2}$$

Így (2.1) és (2.2) összevetésével

$$\beta_m = \frac{\det(\mathbf{A}_{m-1})}{\det(\mathbf{A}_m)} = a_{mm}.$$

(ii) Oldjuk meg az $\mathbf{Ax}+\mathbf{b}=0$ egyenletet *Gauss—Jordan-módszerrel.* Ez elvégezhető úgy, hogy a **b** vektorral kibővítjük az **A** mátrixot, úgy, hogy annak $n+1$-edik oszlopaként írjuk a **b**-t. Használjuk az (1.2) képleteket úgy, hogy ezek közül a második és harmadik sorában állókat $j=n+1$ és $l=n+1$-re is számoljuk. Hagyjuk félbe az (1.2) számolást $i=m-1$-nél, akkor az **A** mátrix $n+1$-edik oszlopának első $m-1$ eleme helyén az

$$\mathbf{A}_{m-1}\mathbf{x}_{m-1}+\mathbf{b}_{m-1} = 0$$

egyenletrendszer megoldása

$$\mathbf{x}_{m-1} = -\mathbf{A}_{m-1}^{-1}\mathbf{b}_{m-1}$$

jelenik meg, ahol $\mathbf{x}_{m-1}$ és $\mathbf{b}_{m-1}$ az $\mathbf{x}$, illetve a $\mathbf{b}$ vektorok első $m-1$ komponensét tartalmazó vektorok.

Az (1.2) számolása közben ugyanolyan műveleteket végzünk az $\mathbf{A}$ mátrix minden $v \geqq m$ indexű oszlopában, mint amilyent végeztünk az $n+1$-edik oszlopban. Ezért, ha az $\mathbf{A}$ m-edik oszlopának első $m-1$ elemét tartalmazó vektort $\mathbf{u}_m$-mel jelöljük, akkor az $i=m-1$-edik lépés befejezése után kapott mátrix m-edik oszlopában az $\mathbf{u}_m$ helyére az $-\mathbf{A}_{m-1}^{-1}\mathbf{u}_m$ vektor kerül. Az m-edik lépésben (1.2) negyedik sora szerint ez szorzódik még a_{mm}-mel, ami viszont (i) szerint β_m-mel azonos. Összevetve az (1.1) formulákkal látható tehát, hogy az (1.2) eljárás az m-edik lépésnél az $\mathbf{A}$ mátrix m-edik oszlopának első $m-1$ eleme helyébe az $\mathbf{r}_m$ vektort írja.

(iii) Az inverz számolása közben a mátrix oszlopai és sorai szimmetrikus szerepet játszanak, ezért (1.1) és (1.2) számolásánál a sorokkal és oszlopokkal ugyanolyan műveleteket kell végezni. Az oszlopok azonosságára érvényes megállapítások tehát a sorokra is igazak, amit a fenti meggondolásokhoz hasonlóan külön is igazolni lehet.

(iv) Nézzük meg mit ad az m-edik lépésben az (1.2) eljárás az $\mathbf{A}$ $m-1$-edrendű bal felső szelete helyébe. Ezt az (1.2) harmadik sora a $k, l=1, \ldots, m-1$ indexek mellett adja meg. Mátrix alakban írva a jobboldal első tagja $\mathbf{A}_{m-1}^{-1}$. A második tag első tényezője — mint azt a (ii) pont levezetése közben látuk —, $-\mathbf{A}_{m-1}^{-1}\mathbf{u}_m$. A második tag második tényezője a (iii) pont szerint $-\beta_m \mathbf{u}_m^T \mathbf{A}_{m-1}^{-1}$. Így az (1.2) eljárás is a $\mathbf{P}_{m-1}$ mátrixot adja az $m-1$-edrendű bal felső szelet helyébe.

Ezzel a tétel bizonyítását befejeztük.

3. A 2.1. tétel következményei

1. KÖVETKEZMÉNY. Mint ismeretes a *Gauss—Jordan-elimináció* nem folytatható, ha valamilyen i-re $a_{ii}=0$ lesz, illetve pontatlan eredményt szolgáltat, ha az a_{ii} elem abszolút értékben kicsivé válik. A most bizonyított tétel értelmében ugyanez a probléma lép fel a *rendszámnöveléses módszer*nél is, ha $\beta_k=0$, vagy ha β_k abszolút értéke kicsi.

2. KÖVETKEZMÉNY. Ha az invertálandó mátrix nem szinguláris, akkor a *Gauss—Jordan-eliminációt főelem kiválasztással végezve*, mindig használható és numerikusan stabil, (lásd [4]).

A *rendszámnöveléses módszer*nél is van lehetőség a „*főelem választás*ra". Az m-edik lépésben ez azt jelenti, hogy kiszámoljuk a β_m mennyiséget az összes m-nél nagyobb indexű sor és oszlop választás mellett és ezek közül kiválasztjuk azt a sort és oszlopot, amelyek mellett a β_m nevezője a legnagyobb lesz. Ezt a sort ill. oszlopot kell felcserélnünk az m-edik sorral ill. oszloppal és ezután folytatjuk a számolást.

A rendszámnövelésnél ez a főelem választás nagyon munkaigényes: sokkal munkaigényesebb, mint a *Gauss—Jordan-módszer*nél. Ezért csak elvi lehetőség, de gyakorlatban nem javasolható.

3. KÖVETKEZMÉNY. Ha az invertálandó mátrix pozitív definit, akkor a *Gauss—Jordan-eliminációs módszer* numerikusan stabil, (lásd [4]). Így az előbbi tétel értelmében a pozitív definit mátrixra a rendszámnöveléses invertáló módszer is numerikusan stabil.

4. Alkalmazások

Az [1]-ben vizsgáltuk a

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = g(x, y, u) \tag{4.1}$$

differenciálegyenlet megoldását négyzeten adott peremfeltételek mellett. Az adott iterációs módszert úgy tekinthetjük, mint a rendszámnöveléses módszer általánosítása nemlineáris egyenletrendszerek megoldására. Ötpontos differenciasémát alkalmazva a (4.1) egyenlet az

$$\mathbf{A}(\mathbf{u})\mathbf{u} = \mathbf{b} \tag{4.2}$$

nemlineáris egyenletrendszerbe megy át. Ha $\frac{\partial g}{\partial u} \geqq 0$, akkor az **A** mátrix adott u_0-nál pozitív definit. A 3. következmény értelmében a (4.2) egyenletrendszerre [1]-ben alkalmazott iterációs módszer numerikusan stabil.

IRODALOM

[1] GERGELY, J., „Numerikus módszerek nemlineáris egyenletrendszerek megoldására", *Alkalmazott Matematikai Lapok* **2** (1976) 127—134.
[2] PHILIPS, G. M. and TAYLOR, P. J., *Theory and Application of Numerical Analysis* (Academic Press, 1973).
[3] TEWARSON, R. P., *Sparse Matrices* (Academic Press, 1973).
[4] WILKINSON, J. H., *The Algebraic Eigenvalue Problem* (Oxford, 1965).
[5] Фаддеев, Д. К. и Фаддеева, В. Н., *Вычислительные методы линейной алгебры* (Москва, 1963).

(Beérkezett: 1976. július 15.)

(Újra beérkezett: 1977. március 16.)

DR. GERGELY JÓZSEF
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST I., ÚRI U. 49.

METHODS FOR INVERSION OF MATRICES

J. GERGELY

The paper proves that the method of *Gauss—Jordans elimination* and the *method of bordering* (see [5]) for inversion of matrices are the same.

Alkalmazott Matematikai Lapok **2** *(1976) 363—367*

SZINGULÁRIS MÁTRIXOK ZÉRUS SAJÁTÉRTÉKEINEK LEVÁLASZTÁSA, MÁTRIXOK JORDAN-FÉLE NORMÁLALAKRA HOZÁSA

VARGA GYULA
Budapest

A dolgozat egy véges eljárást ad meg, amelynek segítségével komplex, nemszimmetrikus szinguláris mátrixok sajátérték problémáját visszavezethetjük alacsonyabb rendű reguláris mátrixokra vonatkozó sajátérték feladatra. A módszer alkalmazható mátrixok Jordan-féle normálalakjának meghatározására is.

1. Bevezetés

Ismeretes, hogy nemszimmetrikus szinguláris mátrixok zérustól különböző sajátértékeinek numerikus meghatározása az általában használatos sajátérték-számítási módszerekkel nehézségekbe ütközik. Az egyes iterációs módszerek minden iterációs lépésben megkövetelik a szóban forgó mátrix szorzatalakra való felbontását, amely szinguláris mátrixokra általában nem megy. Szinguláris *Hessenberg típusú mátrixok*ra a [3] könyv ad egy eljárást alacsonyabb rendű, ugyancsak *Hessenberg típusú mátrix* előállítására, amelyből az eredeti mátrix többi, nemzérus sajátértékei kiszámíthatók. Ha azonban az eredeti szinguláris mátrix nem *Hessenberg típusú*, akkor hasonlósági transzformációkkal *Hessenberg típusú*ra transzformálni — az erre szolgáló eljárások műveletigényessége miatt — nem célszerű. A probléma megkerülése az $\mathbf{A}$ szinguláris mátrix helyett $\mathbf{A}+c\mathbf{E}$ (c alkalmasan megválasztott konstans) sajátértékeinek kiszámításával műveletigényes.

Van olyan módszer is, amellyel egyszeres zérus sajátértékű mátrixok többi sajátértékét ki lehet számítani [2], de ennek a hatásfoka szinguláris mátrixokra gyenge.

A 2. szakaszban egy véges számítástechnikai eljárást adunk meg komplex elemű szinguláris mátrixokhoz a zérus sajátérték multiplicitásának szukcesszív redukálására és olyan reguláris mátrix meghatározására, amelynek a sajátértékei az eredeti mátrix nemzérus sajátértékeivel egyeznek meg. Megmutatjuk továbbá, hogyan alkalmazható a módszer tetszésszerinti komplex elemű mátrix ismert többszörös sajátértékéhez tartozó *Jordan-blokkjai*nak a meghatározására. Az ismertetendő módszer a [2] dolgozatban leírt eljárás továbbfejlesztése. A 3. szakaszban egy példát közlünk egy mátrix ismert többszörös sajátértékéhez tartozó blokkjainak a meghatározására, a 4. szakaszban pedig az eljárás gyakorlati alkalmazásával kapcsolatban teszünk megjegyzéseket. Megemlítjük, hogy a 2. szakaszban ismertetendő módszer az automatizálási elméletben már alkalmazásra került.

2. A módszer ismertetése

Ebben a szakaszban vizsgálni fogjuk azt az eljárást, amely tetszőleges $\mathbf{A} \in \mathbf{C}^{n \times n}$ komplex elemű, r rangú mátrixra az alábbi a)—f) lépések alkalmazását kívánja meg:

a) Hajtsunk végre *Gauss-eliminációt* az **A** mátrixon teljes főelemkiválasztással. Ez megfelel annak, hogy az **A** mátrixot jobbról egy **F** permutációs mátrixszal, balról pedig egy olyan **G** mátrixszal szorozzuk meg, amely permutációs mátrixok és főátlójukban csupa 1-eseket tartalmazó alsó háromszögmátrixok szorzata. Így a $\mathbf{C}=\mathbf{GAF}$ mátrixot kapjuk. A sikeresen végrehajtott eliminációs lépések száma az **A** mátrix r rangját adja meg. Ha **A** rangja $r=n$, akkor az eljárás véget ér, az **A** mátrix nem szinguláris, ha $r<n$, az eljárást a b) lépéssel folytatjuk.

b) **A C** mátrixot visszafelé történő eliminációs lépésekkel és alkalmas sornormálásokkal a

$$\mathbf{H} = \begin{bmatrix} \mathbf{E} & \mathbf{W} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{2.1}$$

Hermite-féle normál alakra hozzuk. Ez megfelel annak, hogy a **C** mátrixot balról egy főátlójában csupa 1-eseket tartalmazó **V** felső háromszögmátrixszal és egy **D**, nemzérus elemekből álló átlós mátrixszal szorozzuk meg:

$$\mathbf{H} = \mathbf{DVGAF}.$$

c) **A** *Gauss-eliminációval* párhuzamosan az eredeti **A** mátrixból képezzük a $\mathbf{B}=\mathbf{F}^{-1}\mathbf{AF}$ mátrixot is. Ez azt jelenti, hogy a **B** mátrix **A**-ból a *Gauss-elimináció* végrehajtásánál alkalmazott oszlopcserékkel és a nekik megfelelő sorcserékkel keletkezett, és így hasonló **A**-hoz.

d) Oldjuk meg az $\mathbf{AX}=\mathbf{0}$ egyenletet, hogy meghatározhassuk az **X** oszlopvektorait, amelyek **A** zérusalterét kifeszítik. **A H** és **B** mátrixok előállításából (mindkét mátrix **A**-tól különböző szorzótényezői nemszinguláris mátrixok) látható, hogy az $\mathbf{AX}=\mathbf{0}$ egyenlet ekvivalens a $\mathbf{HY}=\mathbf{0}$ (vagy a $\mathbf{BY}=\mathbf{0}$) és $\mathbf{X}=\mathbf{FY}$ egyenletekkel. Oldjuk meg a $\mathbf{HY}=\mathbf{0}$ egyenletet az $\mathbf{AX}=\mathbf{0}$ helyett. Az

$$\begin{bmatrix} \mathbf{E} & \mathbf{W} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} -\mathbf{W} \\ \mathbf{E} \end{bmatrix} = \mathbf{0} \tag{2.2}$$

egyenlőségből látható, hogy

$$\mathbf{Y} = \begin{bmatrix} -\mathbf{W} \\ \mathbf{E} \end{bmatrix} \tag{2.3}$$

egy teljes megoldásrendszere a $\mathbf{HY}=\mathbf{0}$ egyenletnek.

e) Helyettesítsük be a kapott megoldást a $\mathbf{BY}=\mathbf{0}$ egyenletbe. E célból particionáljuk a **B** mátrixot a **H**-val azonos módon az r-edik sora és r-edik oszlopa mentén:

$$\mathbf{B} = \begin{bmatrix} \mathbf{P} & \mathbf{Q} \\ \mathbf{R} & \mathbf{S} \end{bmatrix}. \tag{2.4}$$

A behelyettesítést elvégezve a

$$\begin{bmatrix} \mathbf{P} & \mathbf{Q} \\ \mathbf{R} & \mathbf{S} \end{bmatrix} \begin{bmatrix} -\mathbf{W} \\ \mathbf{E} \end{bmatrix} = \begin{bmatrix} -\mathbf{PW}+\mathbf{Q} \\ -\mathbf{RW}+\mathbf{S} \end{bmatrix} = \mathbf{0} \tag{2.5}$$

egyenlőséget kapjuk.

f) Hajtsuk végre ezután a **B** mátrixon a következő hasonlósági transzformációt, amelyet az **Y** segítségével építettünk fel:

$$(\mathbf{A}\sim)\,\mathbf{B} \sim \begin{bmatrix} \mathbf{E} & \mathbf{W} \\ \mathbf{0} & \mathbf{E} \end{bmatrix}\begin{bmatrix} \mathbf{P} & \mathbf{Q} \\ \mathbf{R} & \mathbf{S} \end{bmatrix}\begin{bmatrix} \mathbf{E} & -\mathbf{W} \\ \mathbf{0} & \mathbf{E} \end{bmatrix} =$$

(2.6)

$$= \begin{bmatrix} \mathbf{P}+\mathbf{WR} & -(\mathbf{P}+\mathbf{WR})\mathbf{W}+\mathbf{Q}+\mathbf{WS} \\ \mathbf{R} & -\mathbf{RW}+\mathbf{S} \end{bmatrix} = \begin{bmatrix} \mathbf{P}+\mathbf{WR} & \mathbf{0} \\ \mathbf{R} & \mathbf{0} \end{bmatrix}.$$

(Ugyanis (2.5) miatt $-(\mathbf{P}+\mathbf{WR})\mathbf{W}+\mathbf{Q}+\mathbf{WS}=-\mathbf{PW}+\mathbf{Q}+\mathbf{W}(-\mathbf{RW}+\mathbf{S})=\mathbf{0}$). **A P+WR mátrix** előállításához **P** és **R** a (2.4) alatt, **W** pedig a (2.1) alatt található.

Az eljárás eredményét tömörítő (2.6) összefüggés alapján bebizonyítjuk a következő tételt.

2.1. TÉTEL: Az a)—f) eljárás egyszeri alkalmazásával az **A** mátrix $\lambda=0$ sajátértékének multiplicitása (numerikus számítás szempontjából) annyival csökken, amennyi az **A** mátrix $\lambda=0$ sajátértékéhez tartozó *Jordan-blokk*jainak száma.

Bizonyítás: Az a)—f) algoritmus az **A** mátrixon hasonlósági transzformációt végez. Ezért az **A** karakterisztikus polinomját a (2.6) alapján megadja a

$$\det(\lambda\mathbf{E}-\mathbf{A}) = \lambda^{n-r}\det(\lambda\mathbf{E}-(\mathbf{P}+\mathbf{WR}))$$

kifejtése. Az $n-r$ azonban az **A** mátrix defektusa s ez — mint az az elméletből ismert —, megegyezik a mátrix *Jordan-féle normálalak*jában a $\lambda=0$ sajátértékhez tartozó *Jordan-blokk*ok számával. Ezzel a tétel állítását bebizonyítottuk.

Az eljárás és a most bizonyított tétel további elelemzése mind az elmélet, mind a numerikus alkalmazás szempontjából érdekes eredményekre vezet.

Az a)—f) algoritmust ismételten alkalmazhatjuk az $\mathbf{A}_i$ ($i=1, 2, \ldots$) mátrixokra, ahol $\mathbf{A}_0=\mathbf{A}$ és $\mathbf{A}_i$ az $\mathbf{A}_{i-1}$ mátrixra alkalmazott eljárás eredményeként kapott **P+WR** blokk. Ha az eljárást olyan módon végezzük, hogy az oszlopokra alkalmazott operációkat a teljes n-edrendű mátrixra terjesztjük ki, akkor az eljárás ismételt alkalmazása ismételt hasonlósági transzformációkat hajt végre a „nagy" mátrixon. Az eljárás hívásai mindaddig ismételhetők, míg valamilyen $i=k$ indexre az $\mathbf{A}_k$ mátrix már nemszinguláris, azaz nincs defektusa.

Az egyes hívások nyomán egyre finomodó szerkezetű alsó blokkháromszög mátrixot kapunk: a főátlójában álló **0**-blokkok rendje d_i ($i=1, \ldots, k$), ahol d_i az $\mathbf{A}_{i-1}$ defektusa (azaz a rendje és rangja közötti különbség). E mátrix szerkezetből könnyen látható, hogy az **A** mátrix $\lambda=0$ sajátértékéhez d_i számú lineárisan független, legalább i hosszúságú fővektor választható. A lineárisan független, legalább i hosszúságú fővektorok száma megegyezik a legalább i-edrendű *Jordan-blokk*ok számával. Ilyen módon beláttuk, hogy a 2.1. tétel a következő, általánosabb formában érvényes.

2.2. TÉTEL: Az a)—f) eljárás i-edik alkalmazásával az $\mathbf{A}_{i-1}$ mátrix $\lambda=0$ sajátértékének multiplicitása annyival csökken, amennyi az **A** mátrix $\lambda=0$ sajátértékéhez tartozó, legalább i-edrendű *Jordan-blokk*jainak száma ($i=1, \ldots, k$).

1. KÖVETKEZMÉNY: Ezen erősebb tétel alapján meghatározható az **A** mátrix $\lambda=0$ sajátértékéhez tartozó *Jordan-blokk*jainak szerkezete. Erről a következő

módon készíthetünk táblázatot. Egy négyzetrács egymás alatti sorait rendre az eljárás egyes hívásainak eredménye alapján töltjük ki úgy, hogy az i-edik sorban (balszélről kezdve egymás mellett) annyi négyzetbe teszünk jelet, amennyi a rendszámcsökkenés az i-edik hívás nyomán. Az így megszerkesztett táblázatban egy-egy jelekkel teli oszlop egy-egy *Jordan-blokk*nak felel meg, s az oszlop hossza megadja a *Jordan-blokk* méretét. (A 3. szakaszban ilyen táblázat megszerkesztésére mutatunk példát.)

2. KÖVETKEZMÉNY: Legyen $\lambda=\lambda_p$ az $\mathbf{A}$ mátrix nemzérus sajátértéke. Az eljárást az $\mathbf{A}$ helyett a $\lambda_p\mathbf{E}-\mathbf{A}$ mátrixra alkalmazva meghatározhatjuk az $\mathbf{A}$ mátrix $\lambda=\lambda_p$ sajátértékéhez tartozó *Jordan-blokk*jait. Ilyen módon az eljárás alkalmas tetszőleges $\mathbf{A}\in\mathbf{C}^{n\times n}$ mátrix *Jordan-féle normálalak*jának meghatározására.

3. Egy numerikus példa eredményei

Az alábbi táblázat egy 17×17-es mátrix $\lambda=0$-hoz tartozó *Jordan-blokk*jainak a meghatározásához tartozó számítási menetet szemlélteti:

i	A_{i-1} rendszáma	A rendszámcsökkenés az i-edik hívás nyomán
1	17	4
2	13	3
3	10	2
4	8	1
5	7	1
6	6	0

A négyzetrács táblázat az alábbi:

×	×	×	×
×	×	×	
×	×		
×			
×			

A mátrix *Jordan-féle normálalak*jában a $\lambda=0$ sajátértékhez tehát 4 blokk tartozik, ezeknek a nagysága rendre 5, 3, 2 illetve 1. Az eljárás 5. hívása után kapott 6×6-os $\mathbf{A}_5$ mátrixnak már csak nemzérus sajátértékei vannak.

4. Megjegyzések

a) Egy adott sajátértékhez tartozó blokkstruktúra meghatározásához az eljárást annyiszor kell hívni, amennyi az illető sajátértékhez tartozó legnagyobb *Jordan-blokk* rendszáma. Ha a fővektorokra nincs szükségünk, csupán a blokkstruktúra ismeretére, akkor az eljárást elég az egyre csökkenő rendű $\mathbf{A}_i$ mátrixokra korlátozni, s nem kell az oszlopoperációkat az egész „nagy" mátrixon elvégezni.

b) Az eljárás tényleges végrehajtásának műveletigénye az $\mathbf{A}_i$ mátrixokra végrehajtott *Gauss-elimináció* és a **WR** szorzatmátrix kiszámításának műveletigénye hívásonként összegezve. Ez utóbbi mátrixszorzás műveletigénye a *Winograd-féle skalárszorzat képzési módszer* [4] alkalmazásával csökkenthető; a szokásos mátrixszorzási eljárás alapján elvégzendő $r^2\times(n-r)$ számú szorzás helyett csupán $2r\times(n-r)+r\times(r-2)\times\mathrm{int}\left(\frac{n-r+1}{2}\right)$ számú szorzást igényel.

c) Valós mátrixok esetén a 0 és a valós nemzérus sajátértékekhez tartozó blokkstruktúra meghatározása valós aritmetikával történhet, a komplex sajátértékeké komplex aritmetikával, de figyelembe véve, hogy a *Jordan-féle normálalak*ban a konjugált komplex sajátértékpárok blokkszerkezete megegyezik, minden többszörös konjugált komplex sajátértékpár esetén csak egyszer kell a blokkszámokat és nagyságokat megállapítani.

d) A mátrix *Jordan-féle normálalak*jának létezésével és szerkezetének egyértelműségével nem foglalkozunk. Ezekkel kapcsolatban l. pl. az [1] 4. fejezetét.

IRODALOM

[1] RÓZSA, P., *Lineáris algebra és alkalmazásai* (Műszaki Könyvkiadó, Budapest, 1974).
[2] VARGA, GY., „Mátrixok sajátértékeinek meghatározása a ‚hasonlósági elimináció' módszerével", *MTA SZTAKI Közlemények* 5/1969.
[3] WILKINSON, J. H., *The Algebraic Eigenvalue Problem* (Oxford University Press, London, 1965).
[4] WINOGRAD, S., *A New Algorithm for Inner Product* (IEEE Transactions on Computers, 1968).

(Beérkezett: 1976. október 13.)

(Újra beérkezett: 1977. február 3.)

DR. VARGA GYULA
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST I., ÚRI U. 49.

REDUCTION OF AN EIGENVALUE PROBLEM

GY. VARGA

The paper gives a finite procedure to reduce the eigenvalue problem of complex nonsymmetric singular matrices to the eigenvalue problem of lower order nonsingular matrices. The method can be applied to determine the *Jordanian normal form* of matrices.

Alkalmazott Matematikai Lapok **2** *(1976) 369—375*

KONJUGÁLT IRÁNY MÓDSZEREK HIBABECSLÉSEI

ABAFFY JÓZSEF és GALÁNTAI AURÉL
Budapest

A dolgozatban általános hibabecslést adunk meg a konjugált irányokat használó módszerek osztályára.

1. Bevezetés

A dolgozat célja meghatározni egy általános hibabecslést a konjugált irányokat használó módszerek osztályára. Mint ismeretes, a konjugált irány módszereket általában feltétel nélküli függvényminimalizálások esetén használják, de alkalmazhatók lineáris és nemlineáris egyenletrendszerek megoldására is. Megjegyezzük azonban, hogy lineáris egyenletrendszerek megoldására a konjugált irány módszereket általában nem használják, mert jobb módszerek is ismeretesek.

A konjugált irány módszerek nagy száma és gyakori alkalmazása a függvényminimalizálásban szükségessé teszi egy olyan általános hibabecslés kidolgozását, amelyben egy konkrét változat paraméteres formában jelentkezik. A problémát a következő formába fogalmazzuk meg. Legyen

$$f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c \quad (\mathbf{b}, \mathbf{x} \in R^n,\ c \in R^1), \tag{1.1}$$

$\mathbf{A}$ $n \times n$-es szimmetrikus és pozitív definit mátrix. Határozzuk meg az $f(\mathbf{x})$ függvény $\mathbf{x}^*$ minimumhelyét valamilyen konjugált irány módszerrel a következőképpen.

Legyen $\mathbf{x}_1 \in R^n$ tetszőleges és

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i, \quad i = 1, 2, \ldots, n, \tag{1.2}$$

ahol

$$\alpha_i = -\frac{\mathbf{g}(\mathbf{x}_i)^T \mathbf{d}_i}{\mathbf{d}_i^T \mathbf{A} \mathbf{d}_i} \tag{1.3}$$

és

$$\mathbf{g}(\mathbf{x}_i) = \operatorname{grad} f(\mathbf{x}_i). \tag{1.4}$$

A $\{\mathbf{d}_i\}_1^n$ vektorok a konkrét módszer által meghatározott konjugált irányokat jelölik. Ezekről az általánosság megszorítása nélkül feltehetjük, hogy

$$\|\mathbf{d}_i\| = 1, \quad i = 1, 2, \ldots, n.$$

Megjegyezzük, hogy az α_i értéket nem kvadratikus függvények esetén (függvényminimalizálásoknál) az

$$f(\mathbf{x}_{i+1}) = \min_{\alpha \in R^1} f(\mathbf{x}_i + \alpha \mathbf{d}_i)$$

reláció definiálja, amely kvadratikus esetben az (1.3) kifejezéssel megegyezik.

A fenti (1.2)—(1.4) séma hibáját két speciális esetben M. SACHET és S. KAHNE [2] becsülte kvadratikus függvények esetén. Az első esetben feltételezték, hogy az eljárásnak csak egy lépésében követünk el hibát. A második esetben feltették, hogy az α_i és $\mathbf{d}_i$ kiszámításának hibái egymástól függetlenek. Ily módon igen egyszerű becsléseket vezettek le a hibára.

A 2. fejezetben az (1.2)—(1.4) séma hibaterjedésére adunk zárt formulát.

A 3. fejezetben a 2. fejezet alapján pontos a-priori hibabecslést adunk a számított és a pontos minimumhely eltérésére.

A 4. fejezetben az általános hibabecslést alkalmazzuk a *Fox—Wilkinson—Smith-módszer*re, amelyet számpéldával is illusztrálunk.

2. Zárt kifejezés a hibaterjedésre

Jelölje $\{\mathbf{p}_i\}_{i=1}^n$ $(\mathbf{p}_i \neq \mathbf{0})$ a perturbált konjugált irányokat, az $\mathbf{x}_i$ számított értékét pedig $\mathbf{y}_i$ $(i=1, 2, \ldots, n)$.

Legyen továbbá β_i az α_i számított értéke és $\boldsymbol{\delta}_i = \mathbf{p}_i - \mathbf{d}_i$, amely a $\mathbf{d}_i$ irány hibája. Bontsuk fel β_i-t a

$$\beta_i = \alpha_i + \gamma_i + \varepsilon_i \tag{2.1}$$

alakban, ahol

$$\gamma_i = -\frac{\mathbf{g}^T(\mathbf{y}_i)\mathbf{p}_i}{\mathbf{p}_i^T \mathbf{A} \mathbf{p}_i} - \alpha_i, \quad i = 1, 2, \ldots, n \tag{2.2}$$

az akkumulálódott hiba és ε_i a számítógép kerekítési hibája.

2.1. TÉTEL.

Alkalmazzuk az (1.2)—(1.4) eljárást az

$$f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$$

kvadratikus függvényre és tegyük fel, hogy az első k lépést pontosan számítottuk, azaz

$$\beta_i = \alpha_i, \quad \mathbf{p}_i = \mathbf{d}_i, \quad \mathbf{y}_i = \mathbf{x}_i \quad i = 1, 2, \ldots, k; \quad k < n. \tag{2.3}$$

Akkor

$$\mathbf{y}_j = \mathbf{x}_{k+1} + \sum_{i=k+1}^{j-1} \beta_i \mathbf{p}_i, \quad j = k+1, \ldots, n+1, \tag{2.4}$$

és

$$\gamma_i = \left[\alpha_i \boldsymbol{\delta}_i^T \mathbf{A} \boldsymbol{\delta}_i - 2\alpha_i \boldsymbol{\delta}_i^T \mathbf{A} \mathbf{d}_i - \sum_{j=k+1}^{i-1} \beta_j \boldsymbol{\delta}_j^T \mathbf{A} \mathbf{d}_j - \mathbf{g}^T(\mathbf{y}_i)\boldsymbol{\delta}_i\right] / \mathbf{p}_i^T \mathbf{A} \mathbf{p}_i,$$
$$i = k+1, \ldots, n. \tag{2.5}$$

Bizonyítás. Minthogy a (2.4) állítás nyilvánvaló, csupán a (2.5) kifejezést kell igazolnunk i-re vonatkozó indukcióval. A (2.3) feltételek miatt $\mathbf{y}_{k+1}=\mathbf{x}_{k+1}$. Felhasználva a

$$\mathbf{d}_i^T\mathbf{A}\mathbf{d}_i = \mathbf{p}_i^T\mathbf{A}\mathbf{p}_i - 2\boldsymbol{\delta}_i^T\mathbf{A}\mathbf{p}_i + \boldsymbol{\delta}_i^T\mathbf{A}\boldsymbol{\delta}_i \tag{2.6}$$

azonosságot, azt kapjuk, hogy

$$-\mathbf{g}^T(\mathbf{y}_{k+1})\mathbf{p}_{k+1} = -\mathbf{g}^T(\mathbf{x}_{k+1})(\mathbf{d}_{k+1}+\boldsymbol{\delta}_{k+1}) = \alpha_{k+1}\mathbf{d}_{k+1}^T\mathbf{A}\mathbf{d}_{k+1} - \mathbf{g}^T(\mathbf{x}_{k+1})\boldsymbol{\delta}_{k+1} =$$
$$= \alpha_{k+1}\mathbf{p}_{k+1}^T\mathbf{A}\mathbf{p}_{k+1} + [\alpha_{k+1}\boldsymbol{\delta}_{k+1}^T\mathbf{A}\boldsymbol{\delta}_{k+1} - 2\alpha_{k+1}\boldsymbol{\delta}_{k+1}^T\mathbf{A}\mathbf{p}_{k+1} - \mathbf{g}^T(\mathbf{y}_{k+1})\boldsymbol{\delta}_{k+1}] =$$
$$= (\alpha_{k+1}+\gamma_{k+1})\mathbf{p}_{k+1}^T\mathbf{A}\mathbf{p}_{k+1},$$

amely γ_{k+1}-re bizonyítja az állítást. Tegyük fel most, hogy (2.5) igaz i-re ($1 \leqq i < n$). Akkor

$$\gamma_{i+1} = -\alpha_{i+1} - \Big[\mathbf{g}^T(\mathbf{x}_{k+1})\mathbf{d}_{i+1} + \sum_{j=k+1}^{i} \beta_j(\mathbf{d}_j+\boldsymbol{\delta}_j)^T\mathbf{A}\mathbf{d}_{i+1} + \mathbf{g}^T(\mathbf{y}_{i+1})\boldsymbol{\delta}_{i+1}\Big]/\mathbf{p}_{i+1}^T\mathbf{A}\mathbf{p}_{i+1}.$$

Felhasználva most a

$$\mathbf{g}(\mathbf{x}_{i+1})\mathbf{d}_{i+1} = \Big[\mathbf{g}^T(\mathbf{x}_{k+1}) + \sum_{j=k+1}^{i} \alpha_j\mathbf{d}_j^T\mathbf{A}\Big]\mathbf{d}_{i+1} = \mathbf{g}^T(\mathbf{x}_{k+1})\mathbf{d}_{i+1}$$

egyenlőséget és a (2.6) azonosságot $i+1$-re, kapjuk hogy

$$\gamma_{i+1} = -\alpha_{i+1} + [\alpha_{i+1}\mathbf{p}_{i+1}^T\mathbf{A}\mathbf{p}_{i+1} + \alpha_{i+1}\boldsymbol{\delta}_{i+1}^T\mathbf{A}\boldsymbol{\delta}_{i+1} - 2\alpha_{i+1}\boldsymbol{\delta}_{i+1}^T\mathbf{A}\mathbf{p}_{i+1} -$$
$$- \sum_{j=k+1}^{i} \beta_j\boldsymbol{\delta}_j^T\mathbf{A}\mathbf{d}_{i+1} - \mathbf{g}^T(\mathbf{y}_{i+1})\boldsymbol{\delta}_{i+1}]/\mathbf{p}_{i+1}^T\mathbf{A}\mathbf{p}_{i+1},$$

amivel állításunkat bizonyítottuk.

3. Egy a priori korlát a hibaterjedésre

Az első rész felhasználásával ebben a fejezetben egy korlátot adunk $\|\mathbf{y}_j - \mathbf{x}_j\|$-ra. A becsléshez az $\mathbf{A}$ mátrixra vonatkozó $\lambda_{\max}, \lambda_{\min}$ a priori információra van szükségünk és az aktuálisan használt konjugált irány módszer hibájára.

Szükségünk van a következő lemmára

3.1. LEMMA.

Az (1.1) kvadratikus alakra és az (1.2) sorozatra érvényes a

$$\|\mathbf{g}(\mathbf{x}_i)\| \leqq \|\mathbf{g}(\mathbf{x}_1)\| \sqrt{\lambda_{\max}}/\sqrt{\lambda_{\min}}, \quad i = 1, \ldots, n+1 \tag{3.1}$$

egyenlőtlenség.

Bizonyítás.

Minthogy a főtengely transzformáció a gradiens vektor normáját nem változtatja, a (3.1) kifejezést elegendő bizonyítani az

$$f^*(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i x_i^2 \quad \lambda_1 \geqq \lambda_2 \geqq \ldots \geqq \lambda_n > 0 \tag{3.2}$$

függvényre, amelynek gradiense

$$\mathbf{g}^*(\mathbf{x}) = (2\lambda_1 x_1, \ldots, 2\lambda_n x_n)^T \tag{3.3}$$

Tekintsük a következő nívóhalmazt

$$S(\mathbf{z}) = \{\mathbf{x} \in R^n | f^*(\mathbf{x}) \leqq f^*(\mathbf{z})\}, \quad \mathbf{z} \in R^n, \tag{3.4}$$

és keressük $S(\mathbf{z})$-ben a $\mathbf{g}^*(\mathbf{x})^T\mathbf{g}^*(\mathbf{x})$ maximumát. Felhasználva az $f^*(\mathbf{x}) \leqq K, (K = f^*(\mathbf{z}))$ egyenlőtlenséget, azt kapjuk, hogy

$$\mathbf{g}^*(\mathbf{x})^T\mathbf{g}^*(\mathbf{x}) = 4\sum_{i=1}^{n} \lambda_i^2 x_i^2 \leqq 4\lambda_1\left(K - \sum_{i=2}^{n} \lambda_i x_i^2\right) + 4\sum_{i=2}^{n} \lambda_i^2 x_i^2 =$$

$$= 4\lambda_1 K + 4\sum_{i=2}^{n} \lambda_i(\lambda_i - \lambda_1)x_i^2 \leqq 4\lambda_1 K \leqq 4\frac{\lambda_1}{\lambda_n}\sum_{i=1}^{n} \lambda_i\lambda_n z_i^2 \leqq$$

$$\leqq \frac{\lambda_1}{\lambda_n} 4\sum_{i=1}^{n} \lambda_i^2 z_i^2 = \frac{\lambda_1}{\lambda_n}\|\mathbf{g}^*(\mathbf{z})\|^2,$$

amelyből már következik, hogy

$$\|\mathbf{g}^*(\mathbf{x})\| \leqq \|\mathbf{g}^*(\mathbf{z})\| \sqrt{\lambda_{\max}}/\sqrt{\lambda_{\min}}, \quad \mathbf{x} \in S(\mathbf{z}).$$

Minthogy az (1.2), (1.3) konjugált irány módszerekre

$$f(\mathbf{x}_{i+1}) \leqq f(\mathbf{x}_i), \quad i = 1, 2, \ldots, n+1,$$

így az eljárás nem léphet ki az $S(\mathbf{x}_1)$ nívóhalmazból.

A 3.1 Lemmából következik, hogy

$$\|\mathbf{g}(\mathbf{y}_k)\| \leqq \|\mathbf{g}(\mathbf{x}_1)\| \sqrt{\lambda_{\max}}/\sqrt{\lambda_{\min}}, \quad k > 1, \tag{3.5}$$

ha megköveteljük a meglehetősen formális

$$f(\mathbf{y}_k) \leqq f(\mathbf{x}_1), \quad k > 1$$

feltételt, amely elég kicsi $|\varepsilon_i|$ és $\|\boldsymbol{\delta}_i\|$ értékek esetén teljesül.

A 3.1 Lemma további következménye az

$$|\alpha_i| \leqq \alpha = \|\mathbf{g}(\mathbf{x}_1)\| \sqrt{\lambda_{\max}}/(\sqrt{\lambda_{\min}})^3 \tag{3.6}$$

egyenlőtlenség.

Legyen a maximális kerekítési hiba $\varepsilon = \max_i |\varepsilon_i|$ és a konjugált irányok maximális hibája $\delta = \max_i \|\boldsymbol{\delta}_i\|$. Tegyük fel, hogy

$$\|\mathbf{p}_i\| = 1, \quad i = 1, 2, \ldots, n \tag{3.7}$$

érvényes a számított konjugált irányokra. Akkor a következő tétel igaz.

3.1. TÉTEL.

Tekintsük az (1.2)—(1.3) eljárást és az (1.1) kvadratikus alakot. Tegyük fel $\beta_i = \alpha_i$, $\mathbf{p}_i = \mathbf{d}_i$, $\mathbf{y}_i = \mathbf{x}_i$, $i = 1, 2, \ldots, k$; $k < n$ és

$$f(\mathbf{y}_i) \leqq f(\mathbf{x}_1), \quad i = 1, \ldots, n+1, \tag{3.8}$$

akkor

$$\|\mathbf{y}_j-\mathbf{x}_j\| \leqq (j-k-1)\left[\varepsilon+\frac{\alpha\delta}{\lambda_{\min}}\left(\delta\lambda_{\max}+2\lambda_{\min}+\frac{j-k+2}{2}\|\mathbf{A}\|\right)\right], \tag{3.9}$$

minden $j \geqq k+1$ esetén.

Bizonyítás.

A 3.1 Lemmából és az $\mathbf{x}_j=\mathbf{x}_{k+1}+\sum_{i=k+1}^{j-1}\alpha_i\mathbf{d}_i$ egyenlőségből következik, hogy

$$\|\mathbf{y}_j-\mathbf{x}_j\| \leqq \left\|\sum_{i=k+1}^{j-1}[\alpha_i\delta_i+(\gamma_i+\varepsilon_i)\mathbf{p}_i]\right\| \leqq \sum_{i=k+1}^{j-1}(\alpha\delta+\varepsilon+|\gamma_i|). \tag{3.10}$$

A továbbiakban $|\gamma_i|$-t becsüljük. Minthogy

$$|\gamma_i| \leqq \left[\alpha\delta_i^T\mathbf{A}\delta_i+2\alpha\delta\|\mathbf{A}\|+\|\mathbf{g}^T(\mathbf{y}_i)\|\delta+\sum_{j=k+1}^{i-1}|\beta_j|\delta\|\mathbf{A}\|\right]/\mathbf{p}_i^T\mathbf{A}\mathbf{p}_i$$

és

$$|\beta_i| = |\mathbf{g}^T(\mathbf{y}_i)\mathbf{p}_i/\mathbf{p}_i^T\mathbf{A}\mathbf{p}_i| \leqq \|\mathbf{g}(\mathbf{x}_1)\|\sqrt{\lambda_{\max}}/(\sqrt{\lambda_{\min}})^3,$$

valamint

$$\lambda_{\min}\mathbf{x}^T\mathbf{x} \leqq \mathbf{x}^T\mathbf{A}\mathbf{x} \leqq \lambda_{\max}\mathbf{x}^T\mathbf{x}, \tag{3.11}$$

így $|\gamma_i|$-re kapjuk

$$|\gamma_i| \leqq \alpha\delta[\delta\lambda_{\max}+\lambda_{\min}+(i-k+1)\|\mathbf{A}\|]/\lambda_{\min}, \tag{3.12}$$

és innen (3.9) egyszerű számítással adódik.

Az eredmény azt jelenti, hogy a hibaterjedés lineáris δ-ban. A hibakorlát csökkenthető, ha a $\mathbf{d}_i$ konjugált irányok olyanok, hogy

$$\lambda_{\min} \leqq \mathbf{d}_i^T\mathbf{A}\mathbf{d}_i < \mathbf{d}_{i+1}^T\mathbf{A}\mathbf{d}_{i+1}, \quad i=1, \ldots, n-1. \tag{3.13}$$

Lényegesebb csökkentés azonban csak a kezdeti $\mathbf{x}_1$ vektor alkalmas megválasztásával érhető el a

$$\|\mathbf{g}(\mathbf{x}_1)\| \leqq \|\mathbf{A}\mathbf{x}_1+\mathbf{b}\| \leqq \|\mathbf{b}\| \tag{3.14}$$

feltétel alapján.

Végül megjegyezzük, hogy $\mathbf{A}$ spektrumára vonatkozó Gerschgorin-becsléssel (3.9) $\lambda_{\max}$, $\lambda_{\min}$-től való függése elkerülhető. Az egyszerű $\lambda_{\max} \leqq \|\mathbf{A}\|$ becslés a (3.9) korlát $\lambda_{\max}$-tól való függetlenségét eredményezi.

4. Alkalmazás a Smith módszerre

A fejezetben példát mutatunk a konjugált irányok δ_i hibáinak becslésére a *Smith-módszer* ([3]) esetén.

Ismeretes, hogy a *Smith-módszer* és a *Fox—Wilkinson-módszer* ekvivalens abban az értelemben, hogy a konjugált irányok előállítása azonos [1].

Jelölje $\mathbf{e}_i$ az i-edik koordináta egységvektort és $\mathbf{s}_i$ az i-edik konjugált irányt. Akkor a konjugált irányokat az

$$\mathbf{s}_1 = \mathbf{e}_1,$$

$$\mathbf{s}_i = \mathbf{e}_i - \sum_{j=1}^{i-1} \gamma_{ij}\mathbf{s}_j, \quad i = 2, 3, \ldots, n, \tag{4.1}$$

$$\gamma_{ij} = \frac{\mathbf{e}_i^T \mathbf{A}\mathbf{s}_j}{\mathbf{e}_j^T \mathbf{A}\mathbf{s}_j}, \quad 1 \leqq j \leqq i-1, \quad i = 2, 3, \ldots, n$$

rekurzió definiálja.

Tegyük fel, hogy a (4.1) rekurzió második lépésében σ normájú hibát követtünk el és a további lépések során pontosan számolunk.

A számított konjugált irányokat $\mathbf{s}_i'$-vel, a számított γ_{ij} értékeket γ_{ij}'-vel jelölve fennáll, hogy

$$\mathbf{s}_i = \mathbf{e}_i - \sum_{j=1}^{i-1} \gamma_{ij}'\mathbf{s}_j', \quad i = 2, \ldots, n, \tag{4.2}$$

és

$$\boldsymbol{\delta}_i = \mathbf{s}_i - \mathbf{s}_i' = \sum_{j=1}^{i-1} (\tau_{ij}\mathbf{s}_j - \gamma_{ij}'\boldsymbol{\delta}_j) \tag{4.3}$$

ahol $\tau_{ij} = \gamma_{ij} - \gamma_{ij}'$.

Minthogy

$$\tau_{ij} = \frac{\mathbf{e}_i^T \mathbf{A}\mathbf{s}_j}{\mathbf{e}_j^T \mathbf{A}\mathbf{s}_j} - \frac{\mathbf{e}_i^T \mathbf{A}\mathbf{s}_j'}{\mathbf{e}_j^T \mathbf{A}\mathbf{s}_j'} = \frac{(\mathbf{e}_j^T \mathbf{A}\mathbf{s}_j \mathbf{e}_i^T - \mathbf{e}_i^T \mathbf{A}\mathbf{s}_j \mathbf{e}_j^T)\mathbf{A}\boldsymbol{\delta}_j}{\mathbf{e}_j^T \mathbf{A}\mathbf{s}_j \mathbf{e}_j^T \mathbf{A}\mathbf{s}_j'} \tag{4.4}$$

és $\mathbf{e}_j^T \mathbf{A}\mathbf{s}_j = \mathbf{s}_j^T \mathbf{A}\mathbf{s}_j$, fennáll a

$$|\tau_{ij}| < \frac{\|\mathbf{A}\|^2 \|\boldsymbol{\delta}_j\|}{\lambda_{\min} |\mathbf{e}_j^T \mathbf{A}\mathbf{s}_j'|}$$

egyenlőtlenség. Feltéve, hogy

$$\|\boldsymbol{\delta}_j\| < \frac{\lambda_{\min}}{2\|\mathbf{A}\|} \tag{4.5}$$

kapjuk, hogy $\mathbf{e}_j^T \mathbf{A}\mathbf{s}_j' \geqq \lambda_{\min} - \|\mathbf{A}\| \, \|\boldsymbol{\delta}_j\| > \frac{1}{2}\lambda_{\min}$ és

$$|\tau_{ij}| < \frac{2\|\mathbf{A}\|^2 \|\boldsymbol{\delta}_j\|}{\lambda_{\min}^2}. \tag{4.6}$$

Hasonlóképpen a belátható

$$|\gamma_{ij}'| < \frac{2\|\mathbf{A}\|}{\lambda_{\min}} \tag{4.7}$$

egyenlőtlenség is. A $c = 2\|\mathbf{A}\|(\|\mathbf{A}\| + \lambda_{\min})/\lambda_{\min}^2$ jelöléssel és (4.3) felhasználásával kapjuk, hogy

$$\|\boldsymbol{\delta}_i\| \leqq c\delta[1+c]^{i-3}, \quad i \geqq 3. \tag{4.8}$$

A kapott korlátra az előző fejezetek eredményeit már könnyen alkalmazhatjuk. Az eredmény a *Smith-féle függvényminimalizáló módszer* erős instabilitását mutatja, amit a (4.8) korlátnak megfelelő futási eredmények ([1]) is indokolnak. A hibakorlát ezenkívül azt is mutatja, hogy a háromdimenziós tesztfüggvények nem alkalmasak a hibaterjedési tulajdonságok jelzésére.

IRODALOM

[1] ABAFFY, J., „A feltétel nélküli függvényminimalizálás kvadratikus befejezésű módszerei", *SZTAKI Tanulmányok* 47/1976.
[2] SACHET, M. and KAHNE, S., "Error analysis in conjugate direction methods", sajtó alatt.
[3] SMITH C. S., "The automatic computation of maximum likelihood estimates", *N. C. B. Scientific Dept. Report S. C.* (1962) 846/MR/40.

(Beérkezett: 1976. április 28.)

(Újra beérkezett: 1977. március 16.)

ABAFFY JÓZSEF
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST I., ÚRI U. 49.
GALÁNTAI AURÉL
ELTE TTK NUMERIKUS ÉS GÉPI MATEMATIKA TANSZÉK
1088 BUDAPEST VIII., MÚZEUM KRT 6—8.

ERROR ESTIMATIONS FOR THE CONJUGATE DIRECTION METHODS

J. ABAFFY and A. GALÁNTAI

This paper gives a general error bound for conjugate direction methods assuming the dependence of the errors which appear both in the directions and in the linear minimization.

Alkalmazott Matematikai Lapok 2 (1976) 377—393

ADAPTÍV ELEMEK A LINEÁRIS PROGRAMOZÁSBAN

MAROS ISTVÁN
Budapest

A cikk a folytonos lineáris programozási (LP) feladatok megoldására szolgáló eljárások adaptív vonásaival foglalkozik. Célja elsősorban az, hogy rámutasson az LP eljárások adaptív képességeinek növelését szolgáló kutatások jelentőségére, főbb irányaira. Először foglalkozik a hatékonyság és adaptivitás kérdésével és az adaptivitást mint a hatékonyság kiszolgálóját vizsgálja. Ezután bemutatja a főbb kvantitatív és kvalitatív jellegű adaptív elemeket, esetenként kiemeli ezek jelentőségét, ismertet nyitott kérdéseket és felvet néhány problémát.

Az elég nagy anyag áttekinthető tárgyalása érdekében fel kellett tételezni, hogy az olvasó rendelkezik bizonyos jártassággal az LP-ben. Így meg lehetett takarítani az említett technikák leírását és nagyobb hangsúlyt kapott a címben szereplő adaptivitás.

1. A szimplex eljárás gépi realizációjáról

A lineáris programozási feladat megoldó algoritmusai közül a szimplex módszer bizonyult eddig a leghatékonyabbnak és különféle változataival ma már igen nagyméretű LP feladatokat is sikeresen oldottak meg. DANTZIG szimplex módszerének a publikálása [4] óta igen sok minden történt, míg ez valósággá válhatott. Fejlődtek a megoldás fizikai eszközei, az elektronikus számítógépek is, de ez önmagában véve igen kevés lett volna. A szimplex módszernek olyan változatai fejlődtek ki, amelyek egyre inkább figyelembe veszik és kihasználják a nagyméretű LP feladatok speciális tulajdonságait és igyekeznek a módszer szabad paramétereit úgy megválasztani, hogy a megoldást minél gyorsabban lehessen elérni. Számos érdekes észrevétel, eredmény született ennek a fáradozásnak a kapcsán.

A szimplex módszer gépi realizációja lebegőpontos aritmetikát használ. Ebből fakadóan az aritmetikai műveletek kerekítési hibáktól, jegyveszteségtől terhesek. Érdekes módon nem a multiplikatív, hanem az additív típusú műveletek a „veszélyesek", amikor is a művelet relatív hibája igen nagy lehet és ez a szimplex eljárásra nézve komoly következménnyel járhat [8]. Nagyméretű feladatoknál, ahol igen sok műveletet kell elvégezni a megoldás eléréséig, ez fokozottan igaz.

A nagyméretű LP feladatoknak van azonban egy olyan speciális tulajdonsága, amely sok szempontból (így az említett numerikus szempontból is) előnyösen kihasználható. Ez a tulajdonság pedig az, hogy a nagyméretű LP feladatok feltételi mátrixában a nullától különböző elemek száma az össz elemszámhoz képest kicsi, vagyis az ilyen mátrixok kis kitöltöttségűek, *ritkás*ak. Bár nem szokás definiálni azt, hogy egy feltételi mátrixot mikor nevezünk ritkásnak, az ilyen tulajdonságot általában 10% alatti kitöltöttség esetén lehet említeni. Az igazán nagyobb méretű LP feladatok, melyek 1000-nél is több feltételt tartalmaznak, igen ritkásak, kitöltöttségük általában 0,2% és 1% között van. Sőt, a tapasztalat azt mutatja, hogy

élő feladatok feltételi mátrixai egy bizonyos méreten felül már általában olyanok, hogy oszloponként átlagban 5—10 nullától különböző elemet tartalmaznak. KALAN [7] még azt is észrevette, hogy az amúgy is kevés nem-nulla elem között sok azonos értékű van. Ezt a tulajdonságot *szuper ritkásság*nak nevezzük.

A számítógépeken szokásos számábrázolások közül az egész típusú ábrázolás általában kisebb helyen történik, mint a valós (lebegőpontos) típusú, így adott memória területen több egész számot lehet tárolni, mint valóst. Ha tudjuk azt, hogy egész számaink nem haladnak meg bizonyos értéket (általában valamilyen alacsony 2 hatványt), ez az arány tovább javítható tömörített számábrázolás bevezetésével.

Mindez azért érdekes, mert a ritkás mátrixok tárolásánál csak a nemnulla elemek (valós számok) és azok helymegjelölő adatai (egész számok) kerülnek be a gépbe. Ez azt jelenti, hogy például egy 1%-os kitöltöttségű, 1000×1000-es feladat 1 millió adata helyett elég 10 ezer valós és egész számot tárolni indulásnál. A szuper ritkásság kihasználása esetén pedig, ha például 1000 különböző értékű mátrix bejegyzés van, ezt az 1000 valós értéket és 20 000 egész számot kell esetleg tömörített formában tárolni.

A szimplex módszernek a teljes tabló transzformációs változatánál a ritkásság néhány iteráció alatt megszűnik, a tabló feltelik. Többek között már ez a tulajdonság is alkalmatlanná teszi ezt a változatot nagyobb méretű feladatok megoldására, hiszen például egy 1 millió elemes tabló mozgatása iterációról iterációra a háttér tároló és a központi egység között rendkívül időigényes munka.

A módosított szimplex módszer alkalmazásánál a feltételi mátrixra eredeti állapotban van mindig szükség, ezenkívül kell a mindenkori bázis inverze. Az alapmátrix ritkássága természetesen végig megmarad, a bázis inverze azonban feltelhet. A gépi realizációk mégis szinte kivétel nélkül a módosított szimplex módszert használják. Ennek oka az, hogy az alapmátrix tárolása és mozgatása igen kényelmesen megoldható, ugyanakkor a bázis inverzének a gazdaságos kezelésére egy sor különleges eljárást dolgoztak ki, amelyekkel elérhető, hogy az inverzzel ekvivalens alakzat sűrűsége rossz esetben is alig haladja meg az alapmátrix sűrűségének néhányszorosát.

Az elmúlt időszakban és ma is a kutatás főleg azt célozza, hogy minél nagyobb méretű LP feladatokat és minél gyorsabban, biztonságosabban lehessen megoldani. Az ennek érdekében kifejlesztett eljárások általában több szabad paramétert tartalmaznak, melyek helyes megválasztása nagy mértékben segíti a kitűzött cél elérését. A mai korszerű LP programcsomagoknál a felhasználói paraméterek száma 50—80 körül van. A paraméterek kedvező értéke feladatról feladatra változhat és előre általában nem lehet tudni, hogy mikor milyen választás a célravezető. Ez első hallásra súlyos problémának látszik. Méginkább annak tűnik, ha hozzátesszük, hogy a paraméterek különböző értékeire nem pusztán gyorsabban, vagy lassabban működik az eljárás, hanem akár helytelen választ is adhat a problémára (például lehetséges megoldás létezése esetén azt állapítja meg, hogy nincs lehetséges megoldás), sőt az is előfordulhat, hogy egyáltalán nem kapunk választ (például két invertálás közötti javítást a második invertálás nem reprodukálja, emiatt ciklikusan javítás és visszaesés következik egymás után). Ilyen körülmények között joggal merül fel a kérdés: mi garantálja a programcsomagok megbízhatóságát, illetve az LP-ben kevéssé jártas felhasználónak mekkora esélye van arra, hogy problémáját meg tudja oldani? Szerencsére a helyzet ennyire azért nem sötét. A programcsomagokban majdnem minden paraméternek van előre beállított standard értéke, méghozzá olyan, amely

az átlagos típusú feladatok esetén — a készítők tapasztalata szerint — jól látja el szerepét. A felhasználónak csak abban az esetben kell valamilyen értéket megadni, ha a standard érték nem látszik megfelelőnek.

A sok paraméter szerepe világos: segítségükkel lehet a programot minél jobban „hozzáidomítani" az aktuális feladathoz, hogy azt a lehető leghatékonyabban oldja meg. Ez a lehetőség azonban a helytelen választás veszélyét is magában hordozza, s nem kevés azon esetek száma, amikor gyakorlott felhasználó kezében is sok „üresjáratot" végzett egy-egy korszerű LP programcsomag. Tekintettel arra, hogy az ilyen eseményekről indokolatlanul kevés az irodalmi beszámoló, ezért az van egy kicsit a szakmai köztudatban, hogy az említett korszerű programcsomagok használata már kellő biztonságot jelent bármilyen felhasználó számára. Többnyire magánbeszélegtésekből és saját tapasztalatból azonban olyan kép alakult ki, hogy ez jelenleg még nem igaz és sok „izzadságos" feladatmegoldás történik még napjainkban azért, mert a paraméterek választása nem megfelelő egy-egy feladatra.

Mi lenne az igazi megoldás? Kétségtelenül az, ha lenne egy olyan algoritmus, mely az aktuális feladatot analizálná és megfelelő kritériumok alapján optimálisan választaná meg a szimplex eljárás paramétereit, teljesen tehermentesítve ezáltal a felhasználót és egyúttal megbízható eszközt jelentve bárki kezében. Ez az említett algoritmus jelenleg még nem ismeretes és megalkotása nem is látszik könnyű feladatnak. A megoldás felé vezető úton azonban már több lépés történt és a kutatás ezen a területen erősen élénkül.

2. Hatékonyság, adaptivitás

Az eddigiekben többször is szerepelt a hatékonyság fogalma, így célszerű néhány szóval rávilágítani arra, hogy jelen szövegösszefüggésben mit is értünk ezen. Egy eljárást *hatékony*nak nevezünk, *ha gyors*an *és biztonságos*an működik. A gyorsaságot általában idővel mérjük, de lehet pénzben vagy akár számítógépi erőforrás felhasználásban is kifejezni. A biztonság, más szóval *megbízhatóság* arra utal, hogy az eljárás a gyakorlatban előforduló feladatokra matematikailag *korrekt választ* ad.

Egy LP programcsomaggal dolgozó felhasználó részéről elfogadható az az elvárás, hogy nem sokat, esetleg semmit sem szeretne törődni a programcsomag belső működésével, ugyanakkor megadott feladatára matematikailag egzakt választ vár úgy, hogy ezért például a lehető legkevesebbet kelljen fizetnie, vagy várnia. A paraméterek ilyen célú helyes megválasztásával nem óhajt, vagy nem tud törődni. Ezt a gondot szívesen áthárítja a gépre, illetve a megoldó algoritmusra. Ha egy algoritmus olyan, hogy az adott feladat ismeretében maga állítja be a futását befolyásoló paramétereket, akkor ezt *adaptív algoritmus*nak nevezzük. Célszerű ezt a meghatározást egy kicsit tovább alakítani: egy algoritmust adaptívnak nevezünk, ha a megoldandó feladatot előre, esetleg menet közben elemzi, állandóan értékeli a megoldás menetét és úgy alakítja magát — paramétereinek be-, vagy átállításával —, hogy a végeredményt a lehető leghatékonyabban érje el.

Ez a megfogalmazás már bizonyos optimum-kritériumot is tartalmaz. Ennek az értelmezése azonban nem problémamentes. A hatékonyság két összetevője — gyorsaság, megbízhatóság — egymás ellen dolgozik. A megbízhatóságot többlet munkaráfordítással (például az újrainvertálások segítségével) egy bizonyos határig növelni lehet, ez azonban feltétlenül a gyorsaság rovására megy. Ennek a fordítottja

is igaz általában. Ha például egy eljárás nem törődik a számítások során elkövetett numerikus hibákkal (kerekítési hiba, jegyveszteség), akkor biztos gyorsabban működik, a megbízhatósága azonban lényegesen csökken, amint azt az erre irányuló vizsgálatok egyértelműen mutatták.

Kiindulva abból, hogy az LP programcsomaggal mindenképpen helyes választ akarunk kapni a megadott feladatokra, elfogadhatónak látszik az a megközelítés, hogy a megbízhatóságot elsőbbrendűnek tekintjük a gyorsasággal szemben. Ilyenformán egy adaptív algoritmusnak helyes választ kell adni a feladatokra és ezt a lehető leggyorsabban kell megtennie.

Megjegyezzük, hogy az adaptív algoritmusoknak egyik hagyományos területe a numerikus kvadratúra, de még ott sem sikerült megtalálni „a" legjobb algoritmust.

3. A szimplex eljárás paraméterei

A további tárgyaláshoz szükséges, hogy felírjuk a megoldandó LP feladatot.

$$(\mathbf{I}, \mathbf{A})\begin{pmatrix}\mathbf{v}\\ \mathbf{x}\end{pmatrix} = \mathbf{b} \tag{3.1}$$

részletesebben kiírva:

$$\begin{aligned} v_1 + \quad & a_{11}x_1 + \ldots + a_{1n}x_n = b_1 \\ & \ddots \\ v_m + a_{m1}x_1 + \ldots + a_{mn}x_n &= b_m \end{aligned} \tag{3.2}$$

és az egyes változókra véges, illetve végtelen egyedi alsó (l_j) és felső (u_j) korlátokat lehet megadni. Az **A** mátrix tetszőleges sorát lehet célfüggvénynek tekinteni:

$$\max\,(\text{vagy } \min)\, v_t.$$

A v_i-ket logikai, az x_j-ket strukturális változóknak nevezzük.

Fenti felírás elég általános, mert belefér például a fix értéken rögzített változó (alsó korlát = felső korlát), vagy az előjelben nem korlátozott változó is (végtelen alsó és felső korlát).

A változókat az egyszerűbb hivatkozás kedvéért az alábbi típusokba soroljuk (x_j most logikai és strukturális változót is jelöl egyszerre):

típus	értéktartomány	megjegyzés
0	$l_j = x_j = u_j$	u_j véges
1	$l_j \leqq x_j \leqq u_j$	l_j és u_j véges
2	$l_j \leqq x_j < +\infty$	„hagyományos" LP változó
3	$-\infty < x_j < +\infty$	szabad változó

A szimplex eljárás paraméterei között vannak kvalitatív jellegűek, illetve kvantitatívak. Az alábbiakban — a teljesség igénye nélkül — megemlítünk mindkét csoportból néhány jellemző paramétert.

Kvalitatív paraméterek:

— egy algoritmus kiválasztása egy algoritmus családból (például primál, duál, paraméteres, integer)

— választás induló eljárásokból (például triviális bázis,megadott rész vagy teljes bázis, CRASH technika, ún. „nulladik fázis")

— választás különféle normálási eljárások közül

— inverz ábrázolási módja (például szorzat alak, eliminációs alak)

— optimalizálás iránya (minimalizálás, maximalizálás)

— degenerációs ciklizálás elleni eljárás (például perturbáció, lexikografikus eljárás, relaxáció)

— optimalizálási technika (például legnegatívabb árnyékár, legnagyobb javítás, *Devex* variánsok, többszörös és részleges kiválasztás, *composite* technika).

Kvantitatív paraméterek:

— különféle tűrési paraméterek (például pivot elem választásnál, árnyékáraknál, relatív nullázásnál, abszolút nullázásnál, megoldás pontosságának ellenőrzésénél, inverz pontosságának ellenőrzésénél, pivot választás az invertálás során, degenerációs perturbáció epszilonja)

— stratégiai paraméterek (például *composite* típusú első fázisnál az elsődleges és másodlagos célfüggvény súlya, a többszörös kiválasztásban résztvevő vektorok maximális száma, „minor" iterációk maximális száma, részleges kiválasztás „mettől-meddig" paraméterei, első fázisban a meg-nem-engedettségek súlyozása)

— frekvenciák (például bázis újrainvertálásának gyakorisága, megoldás kimentésére, ellenőrzésére, kiíratására vonatkozó gyakoriságok)

— egyéb paraméterek (például puffer területek méretei, mágneslemezes munkaterületek elosztása, degenerációs ciklizálás figyelésére vonatkozó paraméter, összes iterációk megengedett maximális száma).

Talán ez a korántsem teljes lista is alátámasztja azt, hogy nem feltétlenül könnyű feladat a paraméterek helyes beállítása. (Kicsit tüzetesebb vizsgálódással az is kiderül, hogy például a tűrési paraméterek megválasztásánál ismerni kell még az adott gép lebegőpontos számábrázolási módját, ennek relatív pontosságát is.)

A szimplex algoritmusok korábbi számítógépes realizációi korántsem engedték meg ennyi paraméternek a szabad mozgást, így a felhasználónak kevesebb dolga volt azokkal a programokkal. Igaz, a kapott szolgáltatás színvonala is alacsonyabb volt: a megoldható feladatok méretkorlátai elég alacsonyak voltak, az algoritmusok hatékonysága is viszonylag korlátozott volt. Valójában azokban az eljárásokban is benne voltak implicit módon az imént felsorolt paraméterek, csak rögzítve egy fix értéken és változtathatatlanul. Elsősorban a gyakorlat által felvetett problémák vezettek aztán azokhoz a felismerésekhez, hogy bizonyos helyeken a szimplex módszer eredeti kötöttsége feloldható, egyes részek algebrailag ekvivalens alakokkal helyettesíthetők, továbbá, hogy a nagyméretű LP feladatoknak vannak olyan speciális tulajdonságaik, amelyeket jól ki lehet használni, illetve figyelembe kell venni.

Így jutottunk el odáig, hogy ma már szinte „dúskálunk" a paraméterekben és szinte arról van szó, hogy a jóból is megárt a sok. Időszerűnek látszik egy olyan felügyelő algoritmus megtervezése, amely maga gondoskodik paramétereinek helyes megválasztásáról a korábban vizsgált hatékonyság szempontjainak megfelelően. Ebben az irányban mindenképpen az az első fontos lépés, hogy felmérjük, milyen adaptív lehetőségek kínálkoznak a lineáris programozásban.

4. Adaptív lehetőségek az LP-ben

Az előző pontnak megfelelő csoportosításban fogjuk megvizsgálni a főbb adaptív lehetőségeket:

— válogatás az algoritmusokban,

— numerikus paraméterek szabályozása.

Az elválasztás a gyakorlatban nem ilyen éles, mert egyes numerikus paraméterek értékének bizonyos beállításával minőségileg más algoritmushoz tudunk jutni (például többszörös kiválasztásban egyetlen vektor, vagy részleges kiválasztás az egész mátrixon).

(i) *Válogatás az algoritmusokban*

Az algoritmus szót itt szűkebb és tágabb értelemben egyaránt értjük. Beszélünk tehát szimplex algoritmusról, de azon belül egy algoritmikus technikát mint például a *Devex technikát* szintén algoritmusnak fogunk nevezni. Jelen pontban ez a szóhasználat nem fog kétértelműséget okozni.

Triviális igazság az, hogy válogatni — egyelőre — csak olyan algoritmusok között lehet, amelyek léteznek. Választásra olyankor kerülhet sor, amikor egy funkciónak az ellátására több algoritmus is rendelkezésre áll. Ezek az algoritmusok a végeredményt tekintve elvileg azonosak, azonban hatékonyságuk különböző szituációkban általában eltérő. Az is igaz, hogy ha csak lokális szempontokat veszünk figyelembe és ennek megfelelően választunk egy „legjobb" algoritmust, akkor lehet, hogy ezzel a feladat teljes megoldása szempontjából igen kevéssé hatékony lépést teszünk. Például, ha az első negatív árnyékáru javítóvektort bevonjuk a bázisba, akkor általában igen gyors iterációs lépéseket hajtunk végre, ugyanakkor a megoldás eléréséhez szükséges iterációs lépések száma ilyenkor lényegesen többnek szokott adódni és ez bőven lerontja az egy iterációra eső nagyobb sebesség látszólagos előnyét. Természetesen bonyolultabb példák is adhatók fenti megjegyzés alátámasztására.

A cél tehát az lenne, hogy minden esetben, amikor választ előtt áll az LP optimalizáló eljárás, ki lehessen választani azt az algoritmust, amelyik a feladat teljes megoldása szempontjából a leghatékonyabb.

Korábban már szó volt arról, hogy ilyen felügyelő algoritmus egyelőre nem ismeretes. Jelen elemző cikkben az eddigi fontosabb észrevételeket tárgyaljuk.

a) *Egy szimplex algoritmus kiválasztása egy algoritmus családból*

Ennek a választásnak a megoldás menetének megkezdése előtt van jelentősége. Tekintettel arra, hogy nagyméretű feladatok megoldásánál a teljes tabló transzformációs eljárások gyakorlatilag használhatatlanok a teljes tabló állandó mozgatása miatt, ezért csak a módosított szimplex módszeren nyugvó algoritmusok jöhetnek szóba. A folytonos LP-nél a két alap algoritmus a primál és a duál eljárás. Felhasználók a primál eljárást általában előnyben részesítik, mert — a második fázisban gyakori — lassú konvergencia miatti leállásnál a kapott megengedett megoldás gazdasági értelmezése többnyire egyszerűbb, mint a duál algoritmus által hasonló helyzetben szolgáltatott duál lehetséges megoldásé. Vannak azonban esetek, amikor mégis célszerű a duál algoritmust használni: ha ismeretes egy duál

lehetséges megoldás (például optimalizálás utáni paraméteres vizsgálatok egy részénél), vagy amikor — a (3.2) képlet jelöléseit használva — $m \gg n$ teljesül. Ez utóbbi azért igaz, mert az optimális megoldás eléréséhez szükséges munka (műveletigény) a munkabázis méretének polinomiális függvényével becsülhető és duál algoritmus esetén a munkabázis méretét n definiálja.

A most elmondottak alapján viszonylag könnyű megállapítani kritériumot a primál illetve duál eljárás választására. Bizonyos felhasználói igények miatt azonban célszerű ezt a választást felhasználói paraméter szinten nyitva hagyni oly módon, hogy ha történik rendelkezés valamelyik eljárás aktivizálására, akkor az a rendelkezés lép érvénybe, ennek elmaradása esetén pedig az adaptív választási technika dönt.

b) *Induló eljárások*

Ha egy adott bázisról akarunk indulni, vagy az optimalizálás előtt a „nulladik fázis" elnevezésű elővizsgálatot akarjuk elvégezni, ezt adaptív módon nem lehet felismerni, így külső vezérlés szükséges. Ha ez elmarad, lehet választani a triviális bázis és a CRASH technika, illetve annak variánsai között. Triviális bázisnak a (3.1) képlet szerinti $\mathbf{B}=\mathbf{I}$ (egységmátrix) bázist nevezzük. A CRASH technika alapváltozata egy trianguláris bázist választ ki a hozzá tartozó megoldás lehetségességének figyelése nélkül. Ez egy igen gyors, oszlop transzformáció nélküli eljárás. Használatával még igen erősen degenerált triviális bázismegoldás esetén is általában olyan bázist kapunk, amelyhez tartozó megoldás már nem, vagy alig degenerált, így onnan hatékony szimplex iterációs lépések végezhetők és elmarad a degenerált bázisokon való „vergődés". Noha az eljárásnak van egy szépséghibája (a megengedettséget általában eléggé rontja), erősen degenerált triviális bázismegoldás esetén mégis előnyös a használata.

A CRASH technika variánsai már képesek a megengedettséget és a célfüggvény alakulását is bizonyos mértékig figyelembe venni, természetesen többletmunka árán, de még mindig transzformációk nélkül. Vannak bonyolultabb változatok, amelyek már a feltételi mátrix oszlopait transzformálják és az így adódó helyzetben alkalmazzák a technika valamelyik, már említett változatát.

A CRASH technika tipikusan induló eljárás. Megfelelő változatát használva általában lényegesen kedvezőbb bázist ad, mintha a logikai változókhoz tartozó egységmátrixból indultunk volna. A technika azonban használható olyankor is, amikor a felhasználó által adott bázisjavaslat nem teljes, pontosabban szólva akkor, ha az meglehetősen hiányos. Ilyenkor ugyanis jó esély van a rész-bázis trianguláris résszel történő kiegészítésére.

Meg kell még említeni egy nem közömbös előnyét is a CRASH technikának. Az általa szolgáltatott bázis inverz (miután transzformáció nélkül keletkezik) tömör, vagyis a nem-nulla elemek nem szaporodnak, és egyben pontos is. Ennek következtében a soron következő szimplex iterációs lépések is gyorsak és többnyire pontosak. Mindez pedig igen jól szolgálja a kitűzött hatékonysági célt.

c) *Normálás*

A normálás a lineáris programozás egyik legkevésbé feltárt területe, noha fontosságát senki nem vitatja. E művelet során a feladat elemeinek sor, illetve oszlop normáló faktorokkal való beszorzása (elosztása) történik meg a megoldás kezdete

előtt és a továbbiakban e transzformált feladat megoldására kerül sor. Az eddig elkészült LP programcsomagok többnyire beérték azzal, hogy lehetőséget biztosítottak egy- vagy többféle normálás végrehajtására, beleértve azt az esetet is, amikor a felhasználó adhat meg normáló faktorokat. Ezen túlmenően aztán a felhasználó dolga felmérni azt, hogy a lehetőségeken belül mit tegyen.

A normálással szemben támasztott követelmények tárgyalásánál utalunk TOMLIN [11] cikkére. Eszerint a normálásnak célja lehet, hogy

— a megoldó algoritmus egyszerűbb legyen (például a felső korlátok azonos szintre hozásával)

— a megoldás eléréséhez kevesebb iterációra legyen szükség

— az eljárás numerikus viselkedése javuljon.

Ezekhez hozzá kell még tenni azt, hogy a normálás után kapott mátrix elemeiről tudjuk, hogy nagyjából mekkorák, így bizonyos abszolút tűrési értékek (tárgyalásuk később következik) helyes megválasztásához támpontot kapunk.

A normálás az algoritmus egyszerűsítéséhez fűzött reményeket nem tudja beváltani, amint ezt viszonylag triviális példák is mutatják. Az említett másik két elvárás teljesülése előre még nem látható, így nem ismeretes olyan (optimális?) normáló eljárás, amely az iterációszámot minimalizálja és a numerikus viselkedést is elfogadhatóvá teszi. A numerikus jóság vizsgálatánál elsősorban arról van szó, hogy nagyon különböző nagyságrendű lebegőpontos számokkal végzett additív műveletek során egy kicsi operandus alig, vagy egyáltalán nem járul hozzá az eredményhez, így jelentős jegyveszteségi hibák léphetnek fel, amik a számítások során igen hamar halmozottan jelentkeznek és komoly problémát okozhatnak. Jelenleg a normálás hatékonyságát a kapott elemek nagyságrendjének a terjedelmével, vagy szórásával mérjük. A tapasztalati adatok a szórás csökkentésének a fontosságát támasztják alá.

CURTIS és REID [3] megadta a szórás minimalizálására szolgáló normáló faktorok meghatározásának módját. Módszerük egy $(m+n)\times(m+n)$-es lineáris egyenletrendszer megoldását igényli, ami első hallásra túl nagy feladatnak tűnik. A rendszer speciális tulajdonságait kihasználva azonban egy iterációs eljárásssal 8—10 lépés után már igen jó megoldás nyerhető nagyméretű feladatok esetén is.

A normálásnak ismeretesek egyszerűbb módszerei is, így a legnagyobb abszolút értékű elem (LA) módszere, a geometriai illetve az aritmetikai átlag módszere. Célszerű ezeket a módszereket egymás után többször és keverve alkalmazni, de mindig az LA módszerrel befejezni az eljárást. Így ugyanis azt érjük el, hogy minden sor és oszlop l_∞ normája 1 lesz. (Egy tetszőleges t komponensű $\mathbf{v}$ vektorra az l_∞ norma: $\|\mathbf{v}\| = \max_{1\leq i\leq t} |v_i|$). Ezáltal jó támpontot kapunk az abszolút tűrési paraméterek kezdeti megválasztásához.

Egy adaptív normálási eljárás lehet a következő. Sor és oszlop szerint geometriai normálást végzünk többször egymás után, de legfeljebb egy megadott számszor, illetve előbb abbahagyjuk az eljárást, ha a nem-nulla elemek szórásnégyzete

$$(\sum a_{ij}^2 - (\sum |a_{ij}|)^2/k)/k$$

egy, a gép lebegőpontos számábrázolásának relatív pontosságától függő érték alá nem esik. Itt k-val a teljes mátrix nem-nulla elemeinek a számát jelöltük. Ezután az előzőeknek megfelelően egy LA normálás következik. Ennek a módszernek

a lényege abban áll, hogy figyelembe veszi a használt gép számábrázolási paraméterét.

Tekintettel arra, hogy a normálás kérdései elméletileg még nincsenek maradéktalanul megválaszolva, ezért egy adaptív algoritmustól ma az várható el, hogy képes legyen a felhasználó által szolgáltatott normáló faktorokkal dolgozni, legyen lehetőség normálás nélküli futásra, és végül: egyéb rendelkezés híján lépjen működésbe az imént vázolt adaptív normáló eljárás.

d) *Inverz ábrázolás*

Itt nemcsak az inverz ábrázolásról, hanem az inverzzel végzett műveletek eredményeként adódó vektorok tárolásáról is lesz szó.

Jelen pontban két olyan kérdést vizsgálunk, melyek eldöntése nem lehetséges a futás előtt, hanem a helyes elem kiválasztása a megoldás során történhet folyamatos adaptív módon.

A bázisinverz egyik lehetséges ábrázolási módja a szorzat alak. A mindenkori bázis inverzét elemi transzformációs mátrixok szorzataként írjuk fel: $\mathbf{B}^{-1}= =\mathbf{E}_t\mathbf{E}_{t-1}...\mathbf{E}_1$. Az $\mathbf{E}_i$ mátrixok az egységmátrixtól egyetlen oszlopvektorban térnek el. Ezeket a vektorokat éta vektoroknak nevezzük. A szimplex iterációk során minden egyes báziscsere alkalmával keletkezik egy új éta vektor, így az inverzzel való műveletek egyre több munkát igényelnek és egyre pontatlanabbak lesznek. A szokásos újrainvertálások után a helyzet ismét javul: kevesebb és rövidebb (kisebb sűrűségű) éta vektor reprezentálja a bázisinverzet.

Az invertáló algoritmus rövid éta vektorok készítésére törekszik. Az éta vektorok tárolása tömörített formában történik, ami invertáláskor egyértelműen előny. A szimplex iterációk során keletkező éta vektorok viszont már gyakran nagyon tele vannak, így aránylag kevés iteráció után az éta vektorokat tartalmazó file, az éta file hosszú lesz és ismét újrainvertálásra lesz szükség.

A bázisinverz eliminációs alakja a bázis $\mathbf{B}=\mathbf{LU}$ alakú trianguláris felbontásából indul ki, ahol $\mathbf{L}$ alsó, $\mathbf{U}$ pedig felső trianguláris. Ezután mind az $\mathbf{L}$, mind az $\mathbf{U}$ tényezőt szorzat alakban írja fel, így valójában egy kettős szorzatalak adódik. Ennek a módszernek egyik előnye akkor mutatkozik meg, amikor a bázis kombinatorikusan csak kevéssé triangularizálható, ilyenkor ugyanis a normál szorzatalaknál rövidebb éta-file készíthető. Másik, egyben fontosabb előnye pedig az, hogy lehetséges a triangularitás fenntartása a szimplex iterációk során. Ezzel egyidejűleg az éta-file növekedése sokkal lassúbb, mint a normál szorzatalaknál, így több iterációt lehet csinálni újrainvertálás nélkül. A felsorolt előnyökkel áll szemben a bonyolultabb invertáló algoritmus, a kettős szorzatalakkal végzett műveletek bonyolultabb adminisztrációja és a triangularitás fenntartását biztosító többletmunka.

Az inverz ábrázolás megfelelő formája tehát nem egy előre eldönthető valami, hanem bázisról bázisra változik. Az természetesen nem járható út, hogy egyik lépésben így, a másikban úgy ábrázoljuk az inverzet, hiszen az átállás egy teljes invertálásnyi munkát igényelne, vagy pedig egy darabig minden lépésben mindkét módszer szerinti éta vektorokat el kellene készíteni.

Az esetleg szükséges átállásra jó alkalom kínálkozik a soronkövetkező újrainvertáláskor. Itt az szükséges, hogy jó kritérium álljon rendelkezésre az algoritmus váltás végrehajtására. Jelenlegi ismeretek azt mutatják, hogy az invertálandó bázis

nem-trianguláris részének nagysága ad döntési alapot. A kritikus méret meghatározásánál a várható műveletigényeket kell figyelembe venni. Így tudja az algoritmus hozzáigazítani magát menet közben a megoldandó feladathoz.

A módosított szimplex módszerben az inverzzel végzett műveletek vektorszintűek, vagyis vagy balról szorozzuk az inverzet egy sorvektorral, vagy az inverzzel szorzunk egy oszlopvektort. Felmerül a kérdés: hogyan célszerű tárolni a műveletek közben adódó vektorokat? Explicit alakban, vagy indexesen tömörítve? Ez a kérdés szintén csak a feladat megoldásának alakulása során válaszolható meg. A tömörített indexes tárolás egyik nagy előnye, hogy miután csak a nem-nulla elemeket tartja nyilván, így a gyakori nulla vizsgálat elmaradhat, továbbá ritkás oszlopok esetén ez a tárolási forma gazdaságosabb. Hátránya a bonyolultabb és lassúbb adminisztráció, valamint teltebb vektorok esetén a nagyobb helyigény és lassúbb művelet-végzés. Az explicit tárolás éppen a fordított esetben előnyös. Egyik tárolásról a másikra az áttérés az újrainvertálásnál sokkal gyakoribb főiterációk bármelyikénél lehetséges a futás addigi menetének ismeretében, figyelembe véve a két mód várható műveletigényét.

e) *Degeneráció*

A degeneráció viszonylag gyakori jelenség az LP feladatok megoldása során, az ebből származó ciklizálás azonban nem. Az ilyen ciklizálás ellen van elmélettel alátámasztott biztos védekezés: perturbációs módszer, lexikografikus eljárás. Ezek állandó alkalmazása azonban meglehetősen lassítaná a szimplex iterációkat, ezért bevetésükre csak akkor kerül sor, ha az ugyancsak — éspedig minden degenerált bázis esetén — használt heurisztikus módszerek ellenére ciklizálás gyanúja merül fel.

Itt kétségtelenül a legnehezebb kérdés a ciklizálás tényének a felismerése. Tekintettel arra, hogy az egymás után keletkező bázisok feljegyzése és azoknak a korábbiakkal való folytonos egybevetése igen munkaigényes feladat, ezért ez nem alkalmas a ciklizálás detektálására. Ezért a ciklizálás másik kísérő jelenségét, a célfüggvény értékének „mozdulatlanságát" lehet alapul venni, hiszen ennek figyelése gyakorlatilag nem igényel többletmunkát. Az természetesen igaz, hogy a célfüggvény hosszabb mozdulatlansága még nem feltétlenül jelent ciklizálást, így az ebből levont következtetés esetleg hibás lehet. Az azonban biztos, hogy például a lexikografikus eljárás beléptetése ilyen esetben nem vezet hibás eredményre, legfeljebb a megoldás hatékonysága szenved csorbát. Azért, hogy ez minél kevésbé következzék be, szükséges a megfelelő kritériumot megtalálni. Különösebb elméleti megalapozottság nélkül általában két módszer használata terjedt el. A „lazább" m darab lépést enged meg változatlan célfüggvényérték mellett, míg a „szigorúbb" ugyanerre két invertálás közti lépésszámot engedélyez.

f) *Optimalizálási technika*

Ez az a terület, ahol nagy erőfeszítések történnek a hatékonyság növelésére. Ennek eredménye egy sor algoritmikus technika, amelyek bizonyos helyzetekben előnyösen használhatók. E technikák egy részének jellemzője, hogy van néhány szabad numerikus paraméterük, amelyek be- illetve átállításával a hatékonyságot tovább lehet befolyásolni.

Régóta tapasztalt tény az, hogy a legnegatívabb árnyékáru javító vektor kiválasztása (ez a legegyszerűbb szabály) messze nem a leghatékonyabb, általában

jelentősen több iterációt igényel, mint más módszerek. A legnagyobb javítás módszere ugyan jelentősen redukálja az iterációk számát, de az egy iterációra eső munka oly mértékben növekszik, hogy nagyméretű feladatok esetén ez járhatatlan út.

A két módszer előnyeinek a kihasználásán alapul a többszörös kiválasztás *(multiple pricing)* módszere. Ez különösen a módosított szimplex módszer használata esetén előnyös. Lényege az, hogy a mátrix egyszeri átnézése során kiválasztunk néhányat a legnegatívabb árnyékáru vektorok közül (főiteráció), ezeket betranszformáljuk az aktuális bázisba ($\mathbf{y}=\mathbf{B}^{-1}\mathbf{a}$ típusú művelet) és ezeken a vektorokon a legnagyobb javítás elve alapján iterálunk, amíg lehet. Ezután ismét főiterációs lépésre kerül sor.

Az eljárás verbális leírásából is látható, hogy elég sok „játék"-ra van lehetőség. Itt van először is az, hogy mennyi legyen az az N szám, amennyi negatív árnyékáru vektort akarunk kiválasztani. Ennél fontosabb szempont az, hogy a vektoroknak betranszformált állapotban (indexesen, vagy explicit alakban) el kell férni a központi memóriában, tehát túl sok vektor kiválasztására nincs is mód. Ugyanakkor nem is érdemes túl sok vektort kiválasztani, mert lehet, hogy a munkaigényes betranszformálás után csak néhány vektort tudunk a bázisba bevonni. Ennek a fordítottja is előfordulhat: néhány kiválasztott vektoron túl sok iterációt végezünk, melyek a vége felé már alig, vagy egyáltalán nem javítják a célfüggvényt.

Az iterációk ez esetben ugyanis gyorsak, mert minden szükséges információ a központi memóriában van. Túl sok „apró" iterációt azonban azért nem célszerű csinálni, mert minden báziscsere alkalmával egy új éta vektor is keletkezik és a hosszú éta-file a soron következő főiterációkat lelassítja, ezenkívül az iterációk egyre inkább csak lokális szempontokat vesznek figyelembe, így a teljes megoldás hatékonyságát egyre kevésbé tudják szolgálni. Célszerű tehát az egy főiteráció során végezhető iterációs lépések számát korlátozni, de oly módon, hogy ez a legkevesebb kárt okozza. Egy abszolút korlát felállítása eleve elesik. Helyette adaptív módon célszerű megközelíteni a válaszadást. Lehet például figyelni, hogy az előző főiterációban elért célfüggvény változás (ha ez nem nulla) legalább egy adott hányadát tudja-e produkálni egy tervezett iteráció (ha ez nem első a főiteráción belül), vagy a főiteráció első iterációjában elért változáshoz viszonyítjuk ugyanazt. Nemleges válasz esetén még további szempontokat kell figyelembe venni, ha a tervezett iteráció egy nem-megengedettséget szüntet meg, vagy nulla típusú változót léptet ki a bázisból, akkor mégis célszerű végrehajtani.

A mátrix egyszeri átnézése, főiterációs lépés végzése munkaigényes és lassú eljárás. Ha a mátrix nagyon elnyúlt alakú, vagyis $m \ll n$, akkor jó esély van arra, hogy a mátrix egy részének az átnézése után is megfelelő választék adódik javító vektorokból. Ezt az eljárást részleges kiválasztásnak *(partial pricing)* nevezzük. Ez a gondolat ismét elég sok szabad paramétert ad, amikkel való megfelelő bánás a megoldó algoritmus hatékonyságát növelheti.

Jelöljük K-val azt az indexet, amettől, L pedig azt az indexet, ameddig a mátrixot átnézzük egy főiteráció során. (Most a (3.2)-ben felírt, egységmátrixszal kibővített mátrixot tekintjük, amelynek $m+n$ oszlopa van.) Meg lehet tenni, hogy előre kijelölt pontokkal a mátrixot felosztjuk és a K, L értékpár két szomszédos pontnak felel meg. A mátrixnak csak ezen részét vizsgálva megpróbálunk N darab javító vektort kiválasztani a korábbi elvek szerint (legnegatívabb árnyékáruak). Ha ez nem sikerül, akkor L értékének a következő osztópont indexét adjuk és folytatjuk a keresést. Ha L túlfutna az $m+n$ értéken, akkor ciklikusan elölről vesszük

a mátrixot, szükség esetén egészen $L=K-1$-ig. Az osztópontok menetközbeni újrakijelölését a futás addigi menetének függvényében lehet elvégezni. Egy új főiteráció ott kezdődik, ahol az előzőben a mátrix átnézése befejeződött.

További lehetőségek a részleges kiválasztás stratégiájánál:

— L-lel addig menni, míg egy adott számú (mely N-nél nagyobb) javító vektor választék nem adódik

— L-lel addig menni, amíg nem adódik N darab olyan javító vektor, melynek árnyékára legalább egy bizonyos szintet el nem ér (ami természetesen messzebb van a nullától, mint az optimalitási kritérium).

Természetesen még további, és a feladathoz esetleg jobban alkalmazkodó kritériumokat lehet felállítani.

Minden esetben igaz az, hogy ha valamilyen feltétel nem teljesül, akkor az átnézést akár az egész mátrixra kell kiterjeszteni, hiszen csak így deklarálhatjuk, hogy megoldásnál vagyunk.

Az optimalizálási technikák közt végezetül szólni kell a *Devex* technikáról [6]. Ennek kiemelkedő fontossága van és tulajdonképpen az egyik legjelentősebb vívmány, ami lineáris programozásban az utóbbi évtizedben történt. Azért említjük mégis az optimalizálási technikák végén, mert a tárgyalt többszörös kiválasztás és részleges kiválasztás módszere alkalmazható *Devex*nél, és közös tárgyalásuk így egyszerűsíthető.

Maga a *Devex* technika is egy adaptív eljárásnak tekinthető. Lényege abban áll, hogy módszert ad az árnyékárak normálására, illetve újranormálására minden egyes lépésben. Ennek célja pedig az, hogy a bázisba való belépésnél a célfüggvény változásának szempontját tudja figyelembe venni, így azt a vektort választja ki, amelyiknek a normált árnyékára a legkedvezőbb. Az oszlop-normáló faktorokat — amelyek valójában az oszlopnormák közelítései — minden lépésben módosítja. Ez azonban többletmunkával és a memóriában többlet helyigénnyel jár. A tapasztalat egyértelműen mutatja, hogy ez a többletmunka néhány igen ritkás feladattól eltekintve bőségesen megtérül az iterációszám radikális csökkenése révén. A helyigény kielégítése érdekében esetleg csökkenteni kell a megoldható feladat méretkorlátait.

A *Devex* technika további előnye még a nagyobb numerikus stabilitás és a degenerációval szembeni nagyobb érzéketlenség, ami a kilépő vektor meghatározásánál használt relaxációs technika következménye.

A módszer alkalmazásával szerzett igen kedvező tapasztalatok jelentős ösztönzést adtak a kutatásnak [2], hogy a technika meglevő néhány hátrányos tulajdonságát kiküszöböljék, illetve tisztázzák a kidolgozott *Devex* variánsok előnyös alkalmazásának feltételeit.

A variánsok közül itt csak azt említjük meg, hogy további többlet-memória biztosítása esetén lehetőség van a dinamikus normáló faktorok pontos számítására is az említett közelítés helyett [5].

Szólni kell néhány szót arról, hogy a fentiek alapján mikor célszerű és mikor nem a *Devex* technika alkalmazása. Tekintettel arra, hogy alap esetben is a munka- és a memóriatöbblet n-nel, az oszlopok számával arányos, ezért $m \ll n$ esetén a használat nem előnyös. Más esetben, ha hagyományos technikával túl lassú konvergencia az optimumhoz, vagy ciklizálás gyanúja merül fel, akkor viszont át kell váltani a *Devex*re. A „megéri", „nem éri meg", illetve az „átváltani" kérdést el-

dönteni a futás figyelése és a becsült munkaigény alapján lehet. Erre azonban pontos módszer jelenleg nem ismeretes. Elképzelhető, hogy valószínűségelméleti megközelítésre van szükség itt is, csakúgy, mint néhány korábban említett döntésnél.

(ii) *Numerikus paraméterek szabályozása*

A numerikus paramétereket két csoportba oszthatjuk. Vannak stratégiát és vannak pontosságot szabályozó paraméterek.

a) A stratégiát szabályozó paraméterekről csak röviden szólunk. Ezek egy részéről már volt szó az előzőekben (többszörös és részleges kiválasztás) és arról is, hogy ezek milyen szerepet játszhatnak egy adaptív algoritmusban. Most néhány további paramétert említünk meg.

A bázisinverzet reprezentáló éta file az iterációk során egyre hosszabb és pontatlanabb lesz. A két jelenség egymástól függetlenül is hátrányos. Ennek kiküszöbölésére szolgál a bázisinverz rendszeres újrainvertálása, melynek vezérlése úgy történik, hogy egy előre megadott lépésszám (báziscsere) után automatikusan végrehajtódik az invertálás. Ha ez túl sűrűn történik, akkor kevesebb idő marad a szimplex iterációk végzésére, ha pedig túl ritkán, akkor a nagyon megnövő éta file miatt lelassulnak és pontatlanná válnak az iterációk és ezért lesz lassú, kevéssé hatékony az egész megoldás. Nem könnyű tehát megtalálni azt a pontot, mely a globális hatékonyságot szolgálja.

Akármilyen ügyesen is választjuk meg az újrainvertálás frekvenciáját, ha előre rögzített lépésszámot adunk meg, nem valószínű, hogy ez minden helyzetben jó lesz. Az iterációk két újrainvertálás közt nagyon különbözőek lehetnek. Egyik esetben nagyon stabil bázisokon haladva esetleg még rövid éta vektorok is keletkeznek, s ilyenkor elég lenne viszonylag ritkábban invertálni. Máskor viszont közel szinguláris bázisok miatt sokkal gyakrabban kellene invertálni, amíg az eljárás túl nem jut ezen az instabil helyzeten („rosszul" meghatározott csúcsokon). Tehát az invertálás szükségességét is helyesebb adaptív módon kezelni. Ennek összetevői: az éta-file hossza, az iterációs sebesség alakulása, az inverz pontossága. Ez utóbbiról a pontosságot szabályozó paraméterekről lesz szó.

Bizonyos lépésszám után szokás az aktuális megoldást félretenni, kimenteni, hogy a későbbiekben a megoldást erről az állapotról folytatni, vagy más paraméterekkel megismételni lehessen. Az ilyen kimentéseket rögzített gyakorisággal szokás elvégezni, ahol a lépésszám megválasztásánál két kimentés közti újrafutás idő, vagy költség kihatását lehet figyelembe venni. Néha célszerű soronkívüli kimentésről gondoskodni (például nem korlátos megoldás észlelésénél), de ez a kérdés nem látszik különösebben fontos területnek adaptivitás szempontjából.

A kimentésekhez hasonlóan bizonyos gyakorisággal szokás a megoldás pontosságát közvetlen visszahelyettesítéssel ellenőrizni. Miután ez a művelet feltartja az optimalizálás előrehaladását, ezért gyakori alkalmazása nem célszerű. Valójában ez a lépés is az inverz pontosságának ellenőrzésére szolgál és elsősorban az instabil bázisokon történő iterációk esetén hasznos. A behelyettesítés eredményének kiértékelését a következő pontban tárgyaljuk.

Van még egy fontos paraméter, amelyet a stratégiát szabályozó paraméterek között kell megemlíteni. Amennyiben az első fázisban (lehetséges megoldás keresése) figyelembe kívánjuk venni bizonyos mértékben az igazi v_t célfüggvényt is, akkor ezt bizonyos α súllyal hozzávesszük a meg-nem-engedettségek w mértékéhez

és így a $w+\alpha v_t$-t maximalizáljuk. Ha $\alpha=0$, akkor visszakapjuk a szokásos első fázisú célfüggvényt.

Amennyiben az egész első fázisban $\alpha=0$ választással dolgozunk, akkor előfordulhat, hogy az első megengedett megoldás, melyet az eljárás megtalál, nagyon távol lesz az optimális megoldástól (hiszen az igazi célfüggvény szempontja mindaddig figyelmen kívül maradt). Így az optimum eléréséhez a második fázisban sok lépésre lesz szükség.

Az első fázis elején még általában sok javító vektor van. Ezek különböző (előjeles!) mértékben szolgálják a megengedettséget és az optimumot. Ilyenkor mindenesetre lehetőség van az alternatív javító vektorok közül olyat választani, amely most az összetett célfüggvényt javítja. Az első fázis vége felé a javító vektorok száma lecsökken, és ami w szerint még javít, az $w+\alpha v_t$ szerint már általában nem, ha α változatlan. Ezért v_t súlyát ilyenkor csökkenteni kell, ha szükséges egészen nulláig. Mindenesetre ha az összetett célfüggvény szerint optimumot értünk el és ebben $w\neq 0$, akkor $\alpha=0$-val még egyszer végig kell nézni a mátrixot és ellenőrizni, hogy valóban nincs lehetséges megoldás, illetve van-e lehetségességet javító vektor.

Úgy tűnik, hogy α menetközbeni helyes átállításával elég sokat lehet nyerni, amennyiben első fázist is igénylő futásról van szó.

b) Numerikus pontosságot szabályozó paraméterek.

Minden LP megoldó program kulcskérdése végső soron a pontosság. Korábban már utaltunk arra, hogy a pontatlanság nemcsak a megoldás értékes jegyeit befolyásolja, hanem minőségileg is helytelen eredményhez vezethet. A pontatlanság abból származik, hogy a lebegőpontos aritmetikai műveletek csak véges pontosságúak. Kisebb veszély a kerekítési hiba, nagyobb veszély az esetleg sorozatos jegyveszteség. Ezek „eredményei" számítási szemetek lehetnek az algebrailag helyes nulla helyett, amelyek aztán szignifikáns elemmé előlépve aktív szerephez jutnak a számítás során és a szinguláris bázistól kezdve a legkülönbözőbb problémákat okozhatják [8].

Ez a felismerés az LP programcsomagok készítőit arra késztette, hogy az eljárás több helyén tűrési paramétereket vezessenek be. Ezek úgy működnek, hogy ha egy bizonyos helyzetben egy érték (abszolútértékben) kisebb, mint az ahhoz a helyzethez tartozó tűrési paraméter, akkor az érték nullának minősül. Ez gyakorlatban azt jelenti, hogy pl. egy betranszformált oszlopvektorban ha egy vektorelem abszolút értéke az illető tűrésnél kisebb, akkor helyére nullát ír az algoritmus. Másik szemléletes példa lehet a pivot elem tűrése, ahol még aszerint is különbséget tesznek, hogy szimplex iterációs pivot választásról, vagy invertálási pivotról van szó. Ezek a tűrési értékek a legtöbb LP programcsomagnál még ma is abszolút jellegű tűrések, vagyis a futás előtt megadott és a futás során változatlan értékek. Nagyjából helyes megválasztásukat elősegíti a futást megelőző normálás [10]. A tűrési paraméterek bevezetésével az LP megoldó programok hatásfoka jelentősen javult. A pontosságot szolgáló hatás „melléktermékeként még a gyorsaság is javult, mert a számítási szemetek lenullázása az éta-file hosszára is kedvező hatást gyakorolt, ugyanis rövidebb lett.

A megszaporodott felhasználói paraméterekhez mintegy 8—12 új érték jött hozzá a tűrési paraméterek bevezetésével.

Az abszolút jellegű tűrési paraméterek lényeges hibája, hogy az iterációk során nem képesek az elemek változó nagyságrendjét követni. Így könnyen lenullázhatnak

szignifikáns elemeket és bent hagynak szemeteket, amelyek abszolút értékben nagyok, de az adott helyzetben relative kicsik.

Az elmondottakból rögtön következik, hogy az abszolút nullázásnál jobb eszközre van szükség. A kézenfekvően adódó relatív nullázásnál csak az a kérdés, hogy mihez kell viszonyítani. Az aritmetikai műveletek esetén csak az additív műveletek érdekesek, mert azok relatív hibája lehet nagy. Az LP-ben igen gyakori

$$\bar{a} = a + b$$

műveletnél az eredményt az alábbiak szerint lehet definiálni:

$$\bar{a} = \begin{cases} 0, & \text{ha } |\bar{a}|/|a| < \varepsilon, \quad \varepsilon > 0 \quad \text{kicsi szám} \\ a+b, & \text{különben.} \end{cases}$$

Ennek értelmezése: ha egy kéttagú additív aritmetikai művelet eredményének abszolút értéke sokkal kisebb, mint az első operandus abszolút értéke, akkor nagy a jegyveszteség és az eredményt nullának tekintjük. Itt az ε-nak alapvető szerepe van. Helyes értéke elsősorban a gépi számábrázolás relatív pontosságának a függvénye. Ha a gépi számábrázolás például 7 decimális jegynek felel meg, akkor $\varepsilon = 10^{-5}$ hatására kb. öt értékes jegy elvesztése esetén nullázódik az eredmény. ε csökkenése esetén nagyobb esélyt adunk annak, hogy számítási szemetek bent maradjanak, míg ε növelése esetleg szignifikáns elemeket nulláz le. Ilyen további feltételek mellett kell megválasztani a relatív nullázás tűrésének helyes értékét. Kísérletek [9] azt mutatják, hogy ε elég karakterisztikusan működik, így remény van a választás automatizálására.

A lineáris algebra közelítő módszereinél elért néhány eredmény [12] felhasználása LP-ben jó támpontot ad a relatív tűrési paraméterek egységes szemléletének kialakításához.

Az inverz pontosságának ellenőrzésére minden főiteráció alkalmával jó lehetőség kínálkozik. A BTRAN és az FTRAN [10] operációkkal számított árnyékárak algebrailag megegyeznek. A kétféle kiszámítási mód azonban eltérő numerikus eredményt adhat, ami jellemzi a bázisinverz pontosságát. Az eltérés szignifikáns voltának megállapítására a kiválasztott oszlop normájától függő tűrési paraméter szolgál. Ez a futás minden szakaszában adaptív módon működik.

Az említett megoldás behelyettesítésnél a baloldal és a jobboldal eltérésének szignifikáns volta a sor és a megoldásvektor normájának függvényében állapítható meg.

Azt, hogy egy árnyékár még negatívnak tekintendő-e, a negatív előjelű árnyékárnak az oszlopnormához való viszonya mondja meg hűen. Ez ugyancsak adaptív módon értékelődik ki.

Egy változó értékének a megengedettségét, illetve ennek tűrését szintén dinamikusan, a megoldás vektor normájának függvényében lehet elbírálni.

Fontos kérdés még a pivottűrés dinamikus meghatározása. Itt is az oszlopnorma a támpont, amely minden lépésben változhat. Aktuális értékétől függően a pivottűrés is más és más lehet a különböző oszlopokra, sőt ugyanarra az oszlopra is eltérő lehet két különböző bázis esetén. Az oszlopnormától való függés igényel

még némi meggondolást. Általában a norma valamilyen, a számábrázolási pontosságtól függő hányadát lehet megadni pivottűrésnek.

Ezzel lényegében az összes tűrési paraméter dinamikus megválasztásának a lehetőségét felvázoltuk. Az elvek géptől függetlenül alkalmazhatók. A tűrési paraméterek konkrét megválasztását az algoritmus az adott gép számábrázolási pontosságának függvényében végzi el. Ezt a számábrázolási pontosságot kell csak megadni a számítások kezdetén.

Bármennyire is úgy látszik, hogy a fenti út valóban helyes, az LP programcsomagok fejlesztői nehezen válnak meg az abszolút tűrési paraméterektől. Úgy tűnik, hogy ezeket a fix értékeket tekintik biztos támpontnak, és a toleranciák kezelését nem merik kiengedni a kezükből, hogy azok „önálló életre keljenek". Másik ellenvetés a teljes dinamizálással szemben az lehet, hogy lelassítja a megoldást. Erre nézve az mondható, hogy ez tüzetes vizsgálatot igényel és ahol tényleg igaz, ott esetleg lehet közelítést használni.

E helyen érdemes szólni arról, hogy a numerikus pontosság szabályozásának a kérdése még olyan gépeknél is felmerül, amelyeknél a teljes LP futás dupla pontosságú aritmetikával, 15—20 decimális jegy pontossággal történik (például IBM—370 MPSX). Ez a pontosság feltétlenül a megbízhatóság jelentős javítását eredményezi, azonban elég nagy áron: többlet memóriaigénnyel és lassúbb aritmetikával. Ugyanakkor ezek az LP csomagok az említett tűrési technikákat is használják (abszolútat és relatívat vegyesen).

5. Utószó

Úgy gondoljuk, hogy az adaptivitás szerepét és fontosságát a lineáris programozásban sikerült bemutatni. Az ismertetett adaptív elemek természetesen nem egyforma súllyal szerepelnek a hatékonyság javításában. Az is igaz, hogy nem volt szó minden létező elemről. A cikk azonban nem törekedett — hiszen nem is tudott volna törekedni — teljességre. Célja csupán az volt, hogy beszámoljon az LP rendszerek fejlesztésének egy hasznosnak tűnő irányzatáról.

IRODALOM

[1] BEALE, E. M. L., "The current algorithmic scope of mathematical programming systems", *Math. Progr. Study* **4** (1975).
[2] CROWDER, H. P. and HATTINGH, J. M., "Partially normalized pivot selection in linear programming", *Math. Progr. Study* **4** (1975).
[3] CURTIS, A. R., REID, J. K., "On the automatic scaling of matrices for Gaussian elimination", *J. of the Inst. of Maths. and its Appl.* **10** (1972).
[4] DANTZIG, G. B., "Maximization of a linear function of variables subject to linear inequalities", T. C. Koopmans (szerk.): *Activity analysis of production and allocation* (Wiley, New York, 1951) XXI. fejezet.
[5] GREENBERG, H. J., KALAN, J. E., "An exact update for Harris' TREAD", *Math. Progr. Study* **4** (1975).
[6] HARRIS, P. M. J., "Pivot selection methods of the Devex LP code", *Math. Progr.* **5** (1973).
[7] KALAN, J. E., "Aspects of large-scale in-core linear programming", *Proceedings of the 1971 annual conference of the ACM* (Chicago, 1971).
[8] MAROS, I., „Hibaeljárások lineáris prgramozási algoritmusokban", *Információ-Elektronika* **2** (1974).

[9] MAROS, I. és MÓCSI, J., "Experiences with the dual type GUB algorithm of Grigoriadis", (IX. International Symp. on Math. Progr., Budapest 1976.).
[10] ORCHARD-HAYS, W., *Advanced Linear Programming Computing Techniques* (McGraw, 1968).
[11] TOMLIN, J. A., "On scaling linear programming problems" *Math. Progr. Study* **4** (1975).
[12] WILKINSON, J. H., *The Algebraic Eigenvalue Problem* (Oxford Univ. Press, 1965).

(Beérkezett: 1977. június 15.)

MAROS ISTVÁN
SZÁMKI
1021 BUDAPEST, II. TÁROGATÓ ÚT 110.

ADAPTIVE METHODS IN LINEAR PROGRAMMING

I. MAROS

The paper deals with the adaptive features of continuous linear programming algorithms. Its purpose is to point out the importance and major trends of the research for increasing the adaptive facilities of LP packages. First it deals with the questions of efficiency and adaptivity and considers adaptivity as a tool for efficiency. After that it gives account of the most characteristic adaptive elements of both quantitative and qualitative nature, and presents unsolved problems.

The way of presentation is descriptive and the paper assumes knowledge of algorithmic techniques in linear programming.

Alkalmazott Matematikai Lapok 2 (1976) 395—411

KVADRATIKUS ALAKOK EGY OSZTÁLYÁRÓL

KÉRI GERZSON
Budapest

A dolgozat első hat szakaszában kopozitív mátrixok jellemzésére vonatkozó tételek és ilyenekkel kapcsolatos ellenpéldák kerülnek tárgyalásra. Az itt szereplő eredmények közül bizonyosak már régebben ismertek voltak. (Lásd [2]—[4], [8] és [10]—[13].)

A 7. szakaszban rámutatunk arra, hogy az általános (azaz nem okvetlenül konvex) kvadratikus programozási feladat megfogalmazható egy, mátrixok kopozitivitására vonatkozó, ekvivalens feladat formájában. Ennek alapján megadható az általános kvadratikus programozási feladat egy lehetséges megoldási módjának a váza. Az ezzel kapcsolatos nehézségekre való utalással együtt felvetünk néhány, a nehézségek kiküszöbölésére irányuló problémát.

1. Bevezetés

Korábbi [16] cikkemben bebizonyítottam, hogy ha $n \leqq 3$ és ha $\mathbf{A}$ egy tetszőleges olyan $n \times n$ méretű valós szimmetrikus mátrix, melyre $\mathbf{x}'\mathbf{A}\mathbf{x} \geqq 0$ minden nemnegatív n dimenziós $\mathbf{x}$ vektor esetén, akkor $\mathbf{A}$ felírható egy pozitív szemidefinit valós szimmetrikus mátrix és egy csupa nemnegatív elemből álló valós szimmetrikus mátrix összegeként. Az azóta eltelt időben, nem tudva P. H. DIANANDA [8], A. HORN és M. HALL [10] eredményeiről, sokat gondolkodtam egyrészt a bizonyításnak minden n természetes számra történő kiterjesztésén, másrészt csak az $n=4$ esetre vagy bizonyítás megadásán, vagy ellenpélda konstruálásán.

A továbbhaladás a minimális kopozitív mátrix fogalmának bevezetése és az $n=5$ esetre történő ugrás után sikerült csak. Az $n=5$ esetre talált ellenpélda tagadó választ szolgáltat a probléma kérdésére egyszerre minden $n \geqq 5$ esetén. Később bizonyos típusú mátrixoknak gráfokkal történő reprezentálása segítségével olyan tételekhez jutottam, amelyek lehetővé teszik további ellenpéldák szisztematikus konstruálását.

E dolgozat kéziratának elkészülése után szereztem csak tudomást a [2]—[4], [8] és [10]—[13] cikkek létezéséről, s ekkor tudtam meg, hogy az általam megoldatlan problémáknak tartott kérdések közül bizonyosokra már korábban ismert volt a válasz. Ennek alapján tartozom az alábbiak közlésével kiegészítésképpen.

A dolgozatom 4. szakaszában megfogalmazott 1. és 4. problémára vonatkozóan $n=5$ esetén már [10]-ben szerepelt egy-egy A. HORN ill. MARSHALL HALL által megadott ellenpélda, DIANANDA pedig [8]-ban bebizonyította, hogy az 1. problémában tartalmazott állítás igaz, ha $n \leqq 4$. A szóbanforgó két probléma kapcsolatára vonatkozó alábbi lényegbe vágó eredményt fogalmazta meg HALL és NEWMAN [11]-ben: Az n-edrendű kopozitív mátrixok által alkotott konvex kúp és a nemnegatív együtthatós lineáris alakok négyzetösszegeivel kifejezett n változós kvad-

ratikus alakokhoz tartozó szimmetrikus mátrixok által alkotott konvex kúp egymás polárjai az $n(n+1)/2$ dimenziós euklideszi térben. HALL és NEWMAN eredményének felhasználásával megmutatható, hogy az 1—4. problémák mind ekvivalensek egymással. Ennélfogva $n=4$ esetén már csak az 5. problémára marad kérdéses a válasz.

A dolgozatom 5. szakaszában szereplő tételekhez hasonló tételeket V. BASTON [2]-ben, E. HAYNSWORTH és A. J. HOFFMAN [12]-ben, valamint HOFFMAN és F. PEREIRA [13]-ban bizonyított. Például az 5. szakaszban szereplő 7. tétel megegyezik a [2]-ben található 2.1. lemmával, a 8. tétel pedig hasonlít a [2]-ben található 2.3. tételhez. Az eltérés itt abban áll, hogy BASTON nem a minimális kopozitív mátrixokat, hanem a kopozitív mátrixok kúpjának extremális sugarait vizsgálta. A két eredmény összevetéséből kiderül, hogy a ± 1 elemű mátrixok körében az extrém kopozitív mátrix és a minimális kopozitív mátrix fogalma fedi egymást. (Viszont az $\mathbf{E}_{ij}+\mathbf{E}_{ji}$ mátrix extrém kopozitív, de nem minimális kopozitív, továbbá tetszőleges minimális kopozitív $\mathbf{A}$ mátrix esetén az

$$\begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{A} \end{bmatrix}$$

mátrix minimális kopozitív, de nem extrém kopozitív.) A [13] cikk a ± 1 elemű kopozitív mátrixokra vonatkozó meglevő eredményeket $-1, 0, +1$ elemű mátrixokra általánosítja.

A kopozitív mátrix fogalmát T. S. MOTZKIN vezette be.

2. Alapvető fogalmak és jelölések

Szám alatt mindig valós számot értünk, s ennek megfelelően értelmezzük az euklideszi tér, mátrix, vektor, függvény stb. fogalmakat is.

Egy mátrix vagy vektor elemeire, komponenseire a mátrixot vagy vektort jelölő betűnek megfelelő kisbetűt használjuk, 2 ill. 1 alsó indexszel ellátva. Ha egy mátrixot vagy vektort az elemeivel, komponenseivel adunk meg, akkor szögletes zárójelet használunk. Eszerint, ha $\mathbf{A}$ egy m sorból és n oszlopból álló mátrix (röviden $m\times n$ mátrix), akkor

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \dots a_{1n} \\ a_{21} & a_{22} \dots a_{2n} \\ \dots\dots\dots\dots \\ a_{m1} & a_{m2} \dots a_{mn} \end{bmatrix}.$$

Ha $\mathbf{A}$ egy $n\times n$ mátrix, akkor használjuk még az

$$\mathbf{A} = [a_{ij}]_1^n$$

jelölést is. Általában, ha k_{ij} $(i, j=1, 2, \dots, n)$ skalár értékű kifejezések, akkor

$$[k_{ij}]_1^n$$

az ezekből felépített $n \times n$ mátrixot jelenti. Például

$$[a_{ij}x_ix_j]_1^n = \begin{bmatrix} a_{11}x_1^2 & a_{12}x_1x_2 & \dots & a_{1n}x_1x_n \\ a_{21}x_1x_2 & a_{22}x_2^2 & \dots & a_{2n}x_2x_n \\ \dots & \dots & \dots & \dots \\ a_{n1}x_1x_n & a_{n2}x_2x_n & \dots & a_{nn}x_n^2 \end{bmatrix}.$$

Három fajta egyenlőtlenség jelet használunk, az alábbi értelmezés szerint:
$\mathbf{A} \geqq \mathbf{B}$ (azaz $\mathbf{B} \leqq \mathbf{A}$) ha az $\mathbf{A}-\mathbf{B}$ mátrix minden eleme nemnegatív.
$\mathbf{A} > \mathbf{B}$ (azaz $\mathbf{B} < \mathbf{A}$) ha $\mathbf{A} \geqq \mathbf{B}$ és $\mathbf{A} \neq \mathbf{B}$.
$\mathbf{A} \gg \mathbf{B}$ (azaz $\mathbf{B} \ll \mathbf{A}$) ha az $\mathbf{A}-\mathbf{B}$ mátrix minden eleme pozitív.

Itt $\mathbf{A}$ és $\mathbf{B}$ azonos méretű mátrixok. (A mátrixok fogalmába a vektorokat is beleértjük. Egy vektor lehet vagy oszlopvektor, azaz egy oszlopból álló mátrix, vagy pedig sorvektor, azaz egy sorból álló mátrix.)

A transzponálást vesszővel jelöljük. Vektor esetében vessző nélkül mindig oszlopvektort, vesszővel mindig sorvektort jelölünk.

Speciális vektorok jelölése: $\mathbf{0}$ a csupa 0 elemből álló vektort, $\mathbf{1}$ a csupa 1 elemből álló vektort, $\mathbf{e}_i$ az i-edik alap egységvektort jelenti, azaz olyan vektort, melynek i-edik komponense 1, a többi komponense pedig 0. E vektorok dimenziója mindig kiderül a szövegösszefüggésből.

$\mathbf{E}$-vel egységmátrixot jelölünk, $\mathbf{E}_{ij}$-vel pedig olyan mátrixot, melynek (i, j) eleme 1, a további elemei 0-k

Az n dimenziós euklideszi térre az R^n, ennek nemnegatív ortánsára az R_+^n jelölést használjuk. Tehát

$$R_+^n = \{\mathbf{x}: \mathbf{x} \in R^n, \mathbf{x} \geqq \mathbf{0}\}.$$

3. Pozitív definit, pozitív szemidefinit, kopozitív és minimális kopozitív mátrixok

Néha a pozitív definitség és pozitív szemidefinitség fogalmát egymást kizáró tulajdonságoknak szokták tekinteni. Ezt az értelmezést fogadtam el korábbi [16] cikkemben. Általánosabban elterjedt, és inkább van összhangban más matematikai fogalmak használatával az a másik fajta értelmezés, mely szerint egy pozitív definit kvadratikus alak egyúttal pozitív szemidefinit is. Azért most ezt az utóbbi értelmezést kívánom használni.

1. Definíció. Az $\mathbf{x}'\mathbf{A}\mathbf{x}$ kvadratikus alak pozitív definit, ha $\mathbf{x}'\mathbf{A}\mathbf{x} \leqq 0$-ból következik, hogy $\mathbf{x} = \mathbf{0}$.

2. Definíció. Az $\mathbf{x}'\mathbf{A}\mathbf{x}$ kvadratikus alak pozitív szemidefinit, ha $\mathbf{x}'\mathbf{A}\mathbf{x} \geqq 0$ teljesül minden $\mathbf{x} \in R^n$ esetén.

3. Definíció. Az $\mathbf{x}'\mathbf{A}\mathbf{x}$ kvadratikus alak kopozitív, ha $\mathbf{x}'\mathbf{A}\mathbf{x} \geqq 0$ teljesül minden $\mathbf{x} \in R_+^n$ esetén.

4. Definíció. Az $\mathbf{A}$ mátrix pozitív definit (pozitív szemidefinit, kopozitív), ha $\mathbf{A}$ szimmetrikus mátrix és az $\mathbf{x}'\mathbf{A}\mathbf{x}$ kvadratikus alak pozitív definit (pozitív szemidefinit, kopozitív).

5. DEFINÍCIÓ. Egy szimmetrikus **A** mátrixnak azokat a szubmátrixait, amelyek **A**-nak szimmetrikusan elhelyezkedő elemeiből állnak, az **A** mátrix fő szubmátrixainak, az ezekhez tartozó determinánsokat az **A** mátrix főminorjainak nevezzük.

A fenti definíciók alapján nyilvánvaló, hogy minden pozitív definit mátrix pozitív szemidefinit, minden pozitív szemidefinit mátrix kopozitív, egy pozitív definit (pozitív szemidefinit, ill. kopozitív) mátrixnak minden fő szubmátrixa pozitív definit (pozitív szemidefinit, ill. kopozitív).

6. DEFINÍCIÓ. Az **A** mátrix minimális kopozitív, ha **A** kopozitív, de tetszőleges azonos méretű szimmetrikus **B** mátrixra $\mathbf{B}<\mathbf{A}$ esetén **B** nem kopozitív.

A következő állítások (1—4. tételek) bizonyítása a [16] cikkben megtalálható.

1. TÉTEL. Ha az **A** mátrix kopozitív, és valamely i-re teljesül $a_{ii}=0$, akkor az **A** mátrix i-edik sora (és i-edik oszlopa) csupa nemnegatív elemből áll.

2. TÉTEL. Valamely, a főátlón kívül pozitív elemet nem tartalmazó **A** mátrix akkor és csak akkor kopozitív, ha **A** pozitív szemidefinit.

3. TÉTEL. (A kopozitivitás szükséges és elégséges feltétele.) Egy szimmetrikus **A** mátrix akkor és csak akkor kopozitív, ha az **A** mátrix bármely **B** fő szubmátrixának minden olyan **s** sajátvektorára, melyre $\mathbf{s}\gg\mathbf{0}$, teljesül

$$\mathbf{s}'\mathbf{B}\mathbf{s}\geqq 0.$$

4. TÉTEL. $n\leqq 3$ esetén minden $n\times n$ méretű kopozitív mátrix előállítható egy pozitív szemidefinit mátrix és egy csupa nemnegatív elemből álló szimmetrikus mátrix összegeként.

4. Az alapprobléma és néhány kapcsolódó probléma

A bevezetésben már említett alapprobléma a következőképpen is fogalmazható:

1. PROBLÉMA. Milyen n pozitív egész számok esetén igaz a következő állítás: Minden $n\times n$ méretű kopozitív mátrix előállítható egy pozitív szemidefinit mátrix és egy csupa nemnegatív elemből álló szimmetrikus mátrix összegeként?

A továbbiakban felsoroljuk a kapcsolatos problémákat, a kapcsolatra pedig majd azután mutatunk rá.

2. PROBLÉMA. Milyen n pozitív egész számok esetén igaz a következő állítás: Minden $n\times n$ méretű minimális kopozitív mátrix pozitív szemidefinit?

3. PROBLÉMA. Milyen n pozitív egész számok esetén igaz a következő állítás: Minden $n\times n$ méretű kopozitív **A** mátrix és minden csupa nemnegatív elemből álló $n\times n$ méretű pozitív szemidefinit **Z** mátrix esetén az

$$[a_{ij}z_{ij}]_1^n$$

mátrix kopozitív?

4. PROBLÉMA. Milyen n pozitív egész számok esetén igaz a következő állítás: Minden csupa nemnegatív elemből álló $n\times n$ méretű pozitív szemidefinit **Z** mátrix esetén valamely m-re létezik olyan $m\times n$ méretű **Y** mátrix, melyre $\mathbf{Y}\geqq\mathbf{0}$ és $\mathbf{Y}'\mathbf{Y}=\mathbf{Z}$?

5. PROBLÉMA. Milyen n pozitív egész számok esetén igaz a következő állítás: Bármely n számú, páronként nemnegatív skalárszorzatot adó R^n-beli $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ vektorok esetén található olyan $\mathbf{T}$ unitér mátrix, melyre

$$\mathbf{T}\mathbf{v}_i \geqq \mathbf{0} \quad (i = 1, 2, \ldots, n)?$$

Mielőtt tovább mennék, ezen a helyen mindjárt a választ is megadom, a jelenlegi tudásomnak megfelelően, azaz az $n=4$ eset kizárásával: Az 1—5. problémákban tartalmazott állítások $n \leqq 3$ esetén igazak, $n \geqq 5$ esetén viszont valótlanok. Ennek a kijelentésnek az igazolására a dolgozat további részében folyamatosan sor kerül.

Az alábbi tételből következik, hogy az 1. és a 2. problémában tartalmazott állítások ekvivalensek egymással.

5. TÉTEL. Ha $\mathbf{A}$ egy $n \times n$ méretű kopozitív mátrix, akkor $\mathbf{A}$ előállítható

$$\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2$$

alakban, ahol $\mathbf{A}_1$ minimális kopozitív mátrix, $\mathbf{A}_2$ pedig olyan szimmetrikus mátrix, melyre $\mathbf{A}_2 \geqq 0$.

Bizonyítás. Az

$$\begin{aligned}
&\mathbf{A},\\
&\mathbf{A}_{11} = \mathbf{A} - d_{11}(\mathbf{E}_{11} + \mathbf{E}_{11}),\\
&\mathbf{A}_{12} = \mathbf{A}_{11} - d_{12}(\mathbf{E}_{12} + \mathbf{E}_{21}),\\
&\vdots\\
&\mathbf{A}_{1n} = \mathbf{A}_{1,n-1} - d_{1n}(\mathbf{E}_{1n} + \mathbf{E}_{n1}),\\
&\mathbf{A}_{22} = \mathbf{A}_{1n} - d_{22}(\mathbf{E}_{22} + \mathbf{E}_{22}),\\
&\mathbf{A}_{23} = \mathbf{A}_{22} - d_{23}(\mathbf{E}_{23} + \mathbf{E}_{32}),\\
&\vdots\\
&\mathbf{A}_{2n} = \mathbf{A}_{2,n-1} - d_{2n}(\mathbf{E}_{2n} + \mathbf{E}_{n2}),\\
&\vdots\\
&\mathbf{A}_{n-1,n-1} = \mathbf{A}_{n-1,n-2} - d_{n-1,n-1}(\mathbf{E}_{n-1,n-1} + \mathbf{E}_{n-1,n-1}),\\
&\mathbf{A}_{n-1,n} = \mathbf{A}_{n-1,n-1} - d_{n-1,n}(\mathbf{E}_{n-1,n} + \mathbf{E}_{n,n-1}),\\
&\mathbf{A}_{nn} = \mathbf{A}_{n-1,n} - d_{nn}(\mathbf{E}_{nn} + \mathbf{E}_{nn})
\end{aligned}$$

mátrixok monoton nemnövekvő sorozatot alkotnak a $\geqq$ relációra nézve, ha minden $d_{ij} \geqq 0$. Az n-edrendű kopozitív mátrixok nyilván zárt halmazt alkotnak az $R^{n \times n}$ térben, ezért a d_{ij} számok mindegyike választható úgy, hogy $d_{ij} \geqq 0$ és a fenti mátrix lista minden $\mathbf{A}_{ij}$ eleme kopozitív, de $\mathbf{A}_{ij} - \mu(\mathbf{E}_{ij} + \mathbf{E}_{ji})$ nem kopozitív, ha $\mu > 0$.

Azt állítjuk, hogy a d_{ij} számok ilyen választása esetén az $\mathbf{A}_{nn}$ mátrix minimális kopozitív. Tegyük fel indirekten, hogy van olyan $\mathbf{B} > \mathbf{0}$ szimmetrikus mátrix, melyre $\mathbf{A}_{nn} - \mathbf{B}$ kopozitív. Ekkor valamely i, j párra $i \leqq j$ és $b_{ij} > 0$. Mivel

$$\mathbf{A}_{nn} \leqq \mathbf{A}_{ij}$$

és

$$\mathbf{B} \geqq \frac{1}{2} b_{ij}(\mathbf{E}_{ij} + \mathbf{E}_{ji}),$$

ezért

$$\mathbf{A}_{nn}-\mathbf{B} \leqq \mathbf{A}_{ij}-\frac{1}{2} b_{ij}(\mathbf{E}_{ij}+\mathbf{E}_{ji}).$$

d_{ij} választása miatt

$$\mathbf{A}_{ij}-\frac{1}{2} b_{ij}(\mathbf{E}_{ij}+\mathbf{E}_{ji})$$

nem kopozitív, így

$$\mathbf{A}_{nn}-\mathbf{B}$$

sem lehet kopozitív, ellentétben a feltevéssel.

Az $\mathbf{A}_{nn}$ mátrix tehát minimális kopozitív, melyre $\mathbf{A}_{nn} \leqq \mathbf{A}$. Eszerint $\mathbf{A}_1=\mathbf{A}_{nn}$, $\mathbf{A}_2=\mathbf{A}-\mathbf{A}_{nn}$ a tétel állításának megfelelő mátrixok.

A 3. problémában tartalmazott állítás következik az 1., a 2., a 4. vagy az 5. problémában tartalmazott állítások bármelyikéből, a 4. problémában tartalmazott állítás viszont következik az 5. problémában tartalmazott állításból. Ez egyrészt megmagyarázza, hogyan jutottam a 3—5. problémák felvetéséhez: Az 1. problémában tartalmazott állítás bizonyításának részét képezné a 3. problémában tartalmazott állítás bizonyítása, utóbbit viszont le lehetne vezetni a 4. vagy az 5. problémában tartalmazott állításból, feltéve hogy ezek igazak. A 4. problémában tartalmazott állítás pedig nagyon hasonlít ahhoz a közismerten igaz állításhoz, mely szerint minden pozitív szemidefinit $\mathbf{Z}$ mátrix esetén létezik olyan $\mathbf{Y}$ mátrix, melyre $\mathbf{Y}'\mathbf{Y}=\mathbf{Z}$ vagy más szóval, minden pozitív szemidefinit kvadratikus alak előállítható homogén lineáris funkcionálok négyzetösszegeként. Ennek tudatában azt várhatnánk, hogy minden csupa nemnegatív együtthatót tartalmazó pozitív szemidefinit kvadratikus alak előállítható csupa nemnegatív együtthatót tartalmazó homogén lineáris funkcionálok négyzetösszegeként. A bekezdés elején tett kijelentésből másrészt azt a konklúziót is levonhatjuk, hogy amely n értékekre meg tudjuk cáfolni a 3. problémában tartalmazott állítást, ugyanazokra az n értékekre az 1., a 2., a 4. és az 5. problémában tartalmazott állítás sem igaz.

Az előző bekezdés első mondatában tett kijelentést az alábbiakban bizonyítjuk.

Ismeretes, hogy két pozitív szemidefinit mátrixból elemenkénti szorzással előállított mátrix ugyancsak pozitív szemidefinit. Az viszont nyilvánvaló, hogy két nemnegatív elemű, szimmetrikus mátrixból elemenkénti szorzással előállított mátrix ugyancsak nemnegatív elemű, szimmetrikus mátrix. Ezért, ha az 1. problémában tartalmazott állítás valamely n esetén igaz, akkor a 3. problémában feltételezett tulajdonságú $n \times n$ méretű $\mathbf{A}$ és $\mathbf{Z}$ mátrix esetén

$$\mathbf{A} = \mathbf{B}+\mathbf{C},$$

ahol $\mathbf{B}$ pozitív szemidefinit mátrix, $\mathbf{C}$ csupa nemnegatív elemből álló szimmetrikus mátrix, tehát

$$[a_{ij} z_{ij}]_1^n = [b_{ij} z_{ij}]_1^n + [c_{ij} z_{ij}]_1^n,$$

ahol a jobboldal első tagja pozitív szemidefinit mátrix, második tagja csupa nemnegatív elemből álló szimmetrikus mátrix, a két tag összege tehát kopozitív mátrix. Ezzel beláttuk, hogy az 1. problémában tartalmazott állításból következik a 3. problémában tartalmazott állítás.

Ha valamely n-re a 4. problémában tartalmazott állítást feltételezzük, akkor a 3. problémában feltételezett tulajdonságú $n\times n$ méretű $\mathbf{A}$ és $\mathbf{Z}$ mátrixok esetén tetszőleges $\mathbf{x}\in R_+^n$ vektorra

$$\mathbf{x}'[a_{ij}z_{ij}]_1^n\mathbf{x} = \sum_{i,j=1}^{n} a_{ij}z_{ij}x_ix_j = \sum_{i,j=1}^{n} a_{ij}\left(\sum_{k=1}^{m} y_{ki}y_{kj}\right)x_ix_j = \sum_{k=1}^{m} \mathbf{v}_k'\mathbf{A}\mathbf{v}_k \geqq 0,$$

mert

$$\mathbf{v}_k' = [y_{k1}x_1, y_{k2}x_2, \ldots, y_{kn}x_n] \geqq \mathbf{0}',$$

az $\mathbf{A}$ mátrix pedig kopozitív.

A 4. és az 5. probléma kapcsolatára tett kijelentés bizonyítása: Tudjuk, hogy a 4. problémában feltételezett tulajdonságú tetszőleges $\mathbf{Z}$ mátrixhoz létezik olyan $n\times n$ méretű $\mathbf{V}$ mátrix, melyre $\mathbf{V}'\mathbf{V}=\mathbf{Z}$, továbbá tetszőleges $n\times n$ méretű $\mathbf{V}$ mátrix esetén a $\mathbf{V}'\mathbf{V}$ mátrix pozitív szemidefinit. Ezért a 4. problémában tartalmazott állítás úgy is fogalmazható, hogy ha $\mathbf{V}$ olyan $n\times n$ méretű mátrix, melyre $\mathbf{V}'\mathbf{V}\geqq\mathbf{0}$, akkor valamely m-re létezik olyan $m\times n$ méretű Y mátrix, melyre $\mathbf{Y}\geqq\mathbf{0}$ és $\mathbf{Y}'\mathbf{Y}=\mathbf{V}'\mathbf{V}$. Az 5. problémában tartalmazott állítás viszont úgy is fogalmazható, hogy ha valamely $n\times n$ méretű $\mathbf{V}$ mátrixra $\mathbf{V}'\mathbf{V}\geqq\mathbf{0}$, akkor található olyan $\mathbf{T}$ unitér mátrix, melyre $\mathbf{TV}\geqq\mathbf{0}$. A két átfogalmazott állítás közül az utóbbiból nyilvánvalóan következik az előbbi, hiszen $\mathbf{Y}=\mathbf{TV}$ a kívánt tulajdonságú mátrixot eredményezi.

Az eddigiekből az is kitűnik, hogy ha valamely n esetén a 3. problémában tartalmazott állításra konstruktív cáfolatot, konkrét $\mathbf{A}$ és $\mathbf{Z}$ mátrixszal kifejezett ellenpéldát adunk, akkor ugyanaz az $\mathbf{A}$ mátrix egyúttal ellenpéldát szolgáltat az 1. problémához is, ugyanaz a $\mathbf{Z}$ mátrix ellenpéldát szolgáltat a 4. problémához is, $\mathbf{Z}=\mathbf{V}'\mathbf{V}$ felbontással pedig a $\mathbf{V}$ mátrix oszlopvektorai ellenpéldát szolgáltatnak az 5. problémához. Ha a $\mathbf{Z}$ mátrix páronként ortogonális saját egységvektorai mint oszlopvektorok az $\mathbf{S}$ mátrixot alkotják, a megfelelő sorrendben vett sajátértékek négyzetgyökei az $\mathbf{R}$ diagonálmátrixot alkotják, akkor $\mathbf{v}_i$ gyanánt választhatjuk az $\mathbf{RS}'$ mátrix i-edik oszlopvektorát ($i=1, 2, \ldots, n$).

Szeretném még megemlíteni az 5. problémának a következő szemléletesebb változatát: Milyen n pozitív egész számok esetén igaz, hogy ha tetszőleges adott $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\in R^n$ vektorok közül bármely kettő által kifeszített konvex kúp elforgatás révén elhelyezhető a 2 dimenziós nemnegatív ortánsban, akkor az összes adott vektor által kifeszített konvex kúp elforgatás révén elhelyezhető az n dimenziós nemnegatív ortánsban?

A 4. tétel állítása szerint $n\leqq 3$ esetén igaz az 1. problémában tartalmazott állítás. Ugyancsak $n\leqq 3$ esetén az 5. problémában tartalmazott állítás helyessége elemi úton nagyon könnyen bizonyítható. Ezért a bizonyításra nem térek ki, csak útmutatás gyanánt azt jegyzem meg, hogy $n=2$ esetén az 5. problémában tartalmazott állítás abban az élesebb formában is igaz, amikor az állítást úgy módosítjuk, hogy a $\mathbf{v}_i$ vektorok számát n (azaz 2) helyett ennél nagyobb tetszőleges véges számban állapítjuk meg. (Bizonyítás nélkül megemlítjük, hogy hasonló élesítéssel az állítás nem igaz már $n=3$ esetén sem. Ellenpélda a szabályos oktaéder egy csúcsából kiinduló vektorok. Viszont ugyanez a négy vektor nem szolgáltat ellenpéldát az 5. problémában tartalmazott állításra $n=4$ esetén.)

5. Grafikus mátrixok kopozitivitása

Ebben a szakaszban olyan kvadratikus alakok vizsgálatára szorítkozunk, melyekhez tartozó szimmetrikus mátrix minden eleme $+1$ vagy -1 értékű. A tételek és a bizonyítások leírásánál kényelmi szempontból használunk néhány elemi gráfelméleti fogalmat és tételt, melyeket általánosan ismertnek feltételezünk.

A továbbiakban $\mathbf{A}$ jelentsen egy olyan $n\times n$ méretű szimmetrikus mátrixot, melynek minden eleme $+1$, vagy -1. Feltesszük még, hogy a főátló elemei $+1$-esek, mert ellenkező esetben $\mathbf{A}$ nyilvánvalóan nem lehet kopozitív. Célul tűzzük ki, hogy megadjuk minden ilyen mátrix kopozitivitásának, minimális kopozitivitásának és minden minimális kopozitív ilyen mátrix pozitív szemidefinitségének a szükséges és elégséges feltételét.

Rendeljük hozzá egy tetszőleges ilyen n-edrendű $\mathbf{A}$ mátrixhoz azt a $P_1, P_2, \ldots, P_n$ csúcsú $G_{\mathbf{A}}$ gráfot, melynek P_i és P_j csúcsa akkor és csak akkor van éllel összekötve, ha $a_{ij}=-1$. A fentemlített tulajdonságú mátrixokat ennek alapján nevezhetjük grafikus mátrixoknak:

7. Definíció. Grafikus mátrixnak olyan szimmetrikus mátrixot nevezünk, amelynek minden eleme $+1$ vagy -1 értékű, azonban a főátlóban álló minden eleme $+1$ értékű.

A következő állítás (6. tétel) kimondásával és bizonyításával előkészítjük a grafikus kopozitív mátrixok jellemzésére és osztályozására vonatkozó tételeket (7 - 10. tételek).

6. Tétel. Ha egy grafikus $\mathbf{A}$ mátrixhoz tartozó $G_{\mathbf{A}}$ gráf nem tartalmaz háromszöget (azaz három olyan P_j, P_k, P_l csúcsot, melyek közül bármely kettőt él köt össze), akkor az $\mathbf{A}$ mátrixnak valamely negatív λ sajátértéke esetén

$$(\mathbf{A}-\lambda\mathbf{E})\mathbf{x}=\mathbf{0}$$

$$\mathbf{x}\geqq\mathbf{0}$$

csak úgy teljesülhet, ha $\mathbf{x}=\mathbf{0}$.

Bizonyítás. Ha az $\mathbf{A}$ mátrix valamely sora csupa $+1$ értékű elemből áll, akkor az állítás nyilvánvaló. Ezért feltesszük, hogy $\mathbf{A}$-nak minden sorában van -1 értékű elem. Legyen j egy tetszőleges index. A feltevésünk szerint van olyan k index, hogy $a_{jk}=-1$. Adjuk össze az $(\mathbf{A}-\lambda\mathbf{E})\mathbf{x}=\mathbf{0}$ egyenlőségrendszer j-edik és k-adik sorát:

$$(1)\qquad -\lambda\cdot(x_j+x_k)+\sum_{i\in\{1,2,\ldots,n\}\setminus\{j,k\}}(a_{ji}+a_{ki})x_i=0.$$

Itt az $\mathbf{x}$ vektor minden komponensének együtthatója nemnegatív, mert $a_{ji}+a_{ki}<0$ esetén a $G_{\mathbf{A}}$ gráf tartalmazná a $P_iP_jP_k$ háromszöget. Mivel $\lambda<0$, ezért $\mathbf{x}\geqq\mathbf{0}$ kikötés mellett az (1) egyenlőség csak úgy teljesülhet, hogyha $x_j=0$. Ezzel készen vagyunk a bizonyítással, mert j tetszőleges indexet jelenthetett.

7. Tétel. Egy grafikus $\mathbf{A}$ mátrix akkor és csak akkor kopozitív, ha a $G_{\mathbf{A}}$ gráf nem tartalmaz háromszöget.

Bizonyítás. Ha a $G_{\mathbf{A}}$ gráf tartalmaz egy $P_jP_kP_l$ háromszöget, akkor arra az $\mathbf{x}\in R^n_+$ vektorra, melyre $x_j=x_k=x_l=1$, a többi komponens értéke pedig 0,

$$\mathbf{x}'\mathbf{A}\mathbf{x}=-3$$

adódik, így $\mathbf{A}$ nem lehet kopozitív.

Az **A** mátrix valamely **B** fő szubmátrixához tartozó G_B gráf G_A-nak szubgráfja, ezért, ha a G_A gráf nem tartalmaz háromszöget, akkor G_B sem tartalmaz háromszöget. A 6. tétel szerint ekkor a **B** mátrix tetszőleges negatív λ sajatértéke esetén a hozzá tartozó sajátvektorok között nincs olyan, melynek minden komponense nemnegatív.

A 3. tétel alkalmazásával az eddigiekből adódik, hogy ha a G_A gráf nem tartalmaz háromszöget, akkor az **A** mátrix kopozitív.

8. TÉTEL. Egy grafikus **A** mátrix akkor és csak akkor minimális kopozitív, ha a G_A gráf maximális háromszögmentes (azaz G_A nem tartalmaz háromszöget, de bárhogy kiegészítve egy újabb éllel, az így keletkező gráf már fog tartalmazni háromszöget, vagy más szóval: G_A nem tartalmaz háromszöget, de bármely két csúcsa összeköthető legfeljebb 2 hosszúságú úttal).

Bizonyítás. Ha az **A** mátrix minimális kopozitív, akkor a 7. tétel szerint a G_A gráf háromszögmentes. Ha G_A-t ki lehetne egészíteni egy újabb éllel úgy, hogy háromszögmentes maradjon, akkor az így keletkező gráf egy **A**-nál kisebb kopozitív mátrixhoz tartozna, ezért **A** nem lenne minimális kopozitív.

Fordítva: Ha a G_A gráf maximális háromszögmentes, akkor a 7. tétel szerint **A** kopozitív. Tegyük fel indirekten, hogy G_A maximális háromszögmentes, de **A** nem minimális kopozitív. Ekkor van olyan i, j indexpár, melyre $\mathbf{A}-\varepsilon\cdot(\mathbf{E}_{ij}+\mathbf{E}_{ji})$ kopozitív marad elég kicsi pozitív ε esetén. Maximális háromszögmentes gráf nyilván nem tartalmazhat izolált csúcsot, ezért $i=j$ esetén van olyan k, amelyre $a_{ik}=-1$. Ekkor arra az **x** vektorra, melyre $x_i=x_k=1$, a többi komponens értéke pedig 0, fennáll

$$\mathbf{x}'\left(\mathbf{A}-\varepsilon\cdot(\mathbf{E}_{ij}+\mathbf{E}_{ji})\right)\mathbf{x}=-2\varepsilon<0 \tag{2}$$

minden pozitív ε esetén.

$i\neq j$, $a_{ij}=-1$ esetén vegyük azt az **x** vektort, melyre $x_i=x_j=1$, a többi komponens értéke pedig 0. Ekkor megint fennáll (2) minden pozitív ε esetén.

$i\neq j$, $a_{ij}=1$ esetén van olyan k index, melyre $a_{ik}=a_{jk}=-1$, ellenkező esetben ugyanis a G_A gráfot kiegészítve a P_iP_j éllel, ezáltal háromszögmentes gráfhoz jutnánk. Most arra az **x** vektorra, melyre $x_i=x_j=1$, $x_k=2$, a többi komponens értéke pedig 0, fennáll (2) minden pozitív ε esetén.

Mindenféleképpen ellentmondásra jutottunk, ezért **A** csak minimális kopozitív mátrix lehet.

9. TÉTEL. Egy grafikus **A** mátrix akkor és csak akkor minimális kopozitív és pozitív szemidefinit egyidejűleg, ha G_A teljes páros gráf (vagyis az $\{1, 2, \ldots, n\}$ számhalmazt valamilyen módon két nem üres, közös elemet nem tartalmazó I_1 és I_2 halmazokra lehet bontani úgy, hogy a G_A gráf P_i és P_j csúcsát akkor és csak akkor köti össze él, ha i és j közül egyik az I_1, másik az I_2 halmazban van).

Bizonyítás. Ha G_A teljes páros gráf, akkor G_A nyilvánvalóan maximális háromszögmentes, ezért a 8. tétel szerint **A** minimális kopozitív. **v**-vel jelölve azt az n dimenziós vektort, melyre

$$v_i=\begin{cases}+1, & \text{ha } i\in I_1,\\ -1, & \text{ha } i\in I_2,\end{cases}$$

ekkor fennáll

$$\mathbf{A} = \mathbf{v}\mathbf{v}',$$

tehát **A** pozitív szemidefinit mátrix (és rangja 1).

Most tegyük fel, hogy **A** minimális kopozitív és pozitív szemidefinit. A 8. tétel szerint ekkor G_A maximális háromszögmentes gráf.

Amennyiben G_A páros gráf, akkor G_A teljes páros gráf. Egyébként G_A nem lehetne maximális háromszögmentes gráf.

Az eddigiek alapján elég azt megmutatnunk még, hogy ha G_A maximális háromszögmentes gráf, de G_A nem páros gráf, akkor **A** nem pozitív szemidefinit. Ehhez viszont elég azt megmutatni, hogy a G_A gráfnak van öt olyan

$$P_{k_1}, P_{k_2}, P_{k_3}, P_{k_4}, P_{k_5}$$

csúcsa, hogy a

$$P_{k_1}P_{k_2}, P_{k_2}P_{k_3}, P_{k_3}P_{k_4}, P_{k_4}P_{k_5}, P_{k_5}P_{k_1}$$

éleket G_A tartalmazza, a

$$P_{k_1}P_{k_3}, P_{k_2}P_{k_4}, P_{k_3}P_{k_5}, P_{k_4}P_{k_1}, P_{k_5}P_{k_2}$$

éleket viszont G_A nem tartalmazza. Ekkor ugyanis például a P_{k_1}, P_{k_2}, P_{k_4} csúcsok által meghatározott részgráfhoz tartozó

$$\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{3}$$

mátrix determinánsa negatív értékű.

Tegyük fel tehát, hogy G_A maximális háromszögmentes gráf, de nem páros gráf. Ekkor a G_A gráfban van páratlan hosszúságú kör, és az ilyenek között van minimális hosszúságú is. Egy minimális páratlan hosszúságú $P_{k_1}P_{k_2}\dots P_{k_l}$ kör esetén a P_{k_i}, P_{k_j} csúcsokat nem kötheti össze él, ha i és j nem (ciklikus értelemben) szomszédos indexek. Ha például P_{k_1} és P_{k_j} között futna él, ahol $j \in \{3, 4, \dots, l-1\}$, akkor vagy $P_{k_1}P_{k_2}\dots P_{k_j}$, vagy pedig $P_{k_j}P_{k_{j+1}}\dots P_{k_l}P_{k_1}$ egy rövidebb páratlan hosszúságú kör lenne. $l \neq 3$, mert G_A háromszögmentes. $l < 7$, mert ellenkező esetben G_A maximális háromszögmentes volta miatt a P_{k_1} és P_{k_4} csúcsokat összekötné egy 2 hosszúságú $P_{k_1}QP_{k_4}$ út, és ekkor $P_{k_1}P_{k_2}P_{k_3}P_{k_4}Q$ egy rövidebb páratlan hosszúságú kör lenne. Ezek szerint $l = 5$, s ekkor az **A** mátrix valamely fő szubmátrixa azonos a (3) alatti mátrixszal, így **A** nem pozitív szemidefinit.

A 9. tétel fenti bizonyítása során egyúttal bebizonyítottuk a következőt is:

10. TÉTEL. Egy minimális kopozitív grafikus **A** mátrix akkor és csak akkor indefinit, ha G_A tartalmaz olyan 5 hosszúságú kört, melynek csúcsait a kört kijelölő öt élen kívül további él nem köti össze egymással.

6. Ellenpéldák az n = 5 esetre

Ellenpéldával fogjuk megmutatni, hogy $n = 5$ esetén az 1—5. problémákban tartalmazott állítások közül egyik állítás sem igaz. Ebből már következik, hogy az 1—4. problémákban tartalmazott állítások 5-nél nagyobb n értékek esetén sem igazak. (Egészítsük ki csupa zérót tartalmazó sorokkal és oszlopokkal az $n = 5$

értékhez tartozó ellenpéldákban megadott mátrixokat.) Ha pedig tudjuk, hogy a 3. problémában tartalmazott állítás tetszőleges $n \geqq 5$ esetén hamis, akkor a 4. szakaszban tett megfontolások szerint ebből következik, hogy az 1—5. problémákban tartalmazott állítások közül egyik sem igaz tetszőleges $n \geqq 5$ esetén.

A 8. tétel és a 10. tétel alapján tetszőleges $n \geqq 5$ esetén konstruálni tudunk grafikus mátrixszal megadható ellenpéldát a 2. problémában tartalmazott állításra, s az ilyenek egyúttal ellenpéldát szolgáltatnak az 1. problémában tartalmazott állításra is. Készítsünk el ugyanis egy n csúcsú gráfot a következő algoritmussal:

Az üres gráfból kiindulva, rajzoljunk be először egy 5 hosszúságú kört. Ha ez megtörtént, akkor ismételten ellenőrizzük a gráf csúcspárjait. Mindaddig, amíg találunk két olyan csúcsot, amelyek nincsenek összekötve legfeljebb 2 hosszúságú úttal, akkor húzzuk be az ezeket összekötő élt. Ilymódon nyilvánvalóan véges számú lépés után maximális háromszögmentes gráfhoz jutunk. A megfelelő $n \times n$ méretű grafikus mátrix indefinit minimális kopozitív mátrix.

A fenti konstrukció szerint adódó legegyszerűbb ellenpélda az 1. és a 2. problémában tartalmazott állításra az

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{bmatrix}$$

mátrix, melyhez tartozó $G_{\mathbf{A}}$ gráf nyilván maximális háromszögmentes gráf, de nem páros gráf.

A 3. problémára vonatkozó ellenpélda megadásához tekintsük először az

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

mátrixot. Most legyen $\mathbf{A}$ az előbbi ellenpélda mátrixa, és legyen $\mathbf{Z}=\mathbf{E}+\alpha\mathbf{F}$, ahol α tetszőleges, az $\frac{1}{2}<\alpha \leqq \frac{\sqrt{5}-1}{2}$ egyenlőtlenséget kielégítő valós szám. Az $\mathbf{F}$ mátrix karakterisztikus polinoma $\lambda^5-5\lambda^3+5\lambda-2$. Ennek legkisebb gyöke $-\frac{1}{2}-\frac{\sqrt{5}}{2}$-vel, legnagyobb gyöke pedig 2-vel egyenlő. Következésképpen a $\mathbf{Z}=\mathbf{E}+\alpha\mathbf{F}$ mátrix legkisebb sajátértéke

$$1+\alpha\left(-\frac{1}{2}-\frac{\sqrt{5}}{2}\right) = 1-\frac{\alpha}{\frac{\sqrt{5}}{2}-\frac{1}{2}} \geqq 0,$$

míg az $[a_{ij}z_{ij}]_1^5=\mathbf{E}-\alpha\mathbf{F}$ mátrix legkisebb sajátértéke

$$1-2\alpha < 0.$$

Ebből látható, hogy a **Z** mátrix csupa nemnegatív elemből álló pozitív szemidefinit mátrix. Az **A** mátrixról tudjuk, hogy kopozitív, az $[a_{ij}z_{ij}]_1^5$ mátrix viszont a 2. tétel szerint nem kopozitív.

A 4. szakaszban tett megfontolások szerint a fent meghatározott **Z** mátrix ellenpéldát szolgáltat a 4. problémára vonatkozóan is. A **Z** mátrix sajátértékeinek és sajátvektorainak meghatározása után pedig az 5. problémára vonatkozóan az alábbi ellenpéldát kapjuk:

$$\mathbf{v}_k = \begin{bmatrix} \left(\frac{2\alpha+1}{5}\right)^{1/2} \\ \left(\frac{2+\alpha(\sqrt{5}-1)}{5}\right)^{1/2} \cos\frac{2(k-1)\pi}{5} \\ \left(\frac{2+\alpha(\sqrt{5}-1)}{5}\right)^{1/2} \sin\frac{2(k-1)\pi}{5} \\ \left(\frac{2-\alpha(\sqrt{5}+1)}{5}\right)^{1/2} \cos\frac{4(k-1)\pi}{5} \\ \left(\frac{2-\alpha(\sqrt{5}+1)}{5}\right)^{1/2} \sin\frac{4(k-1)\pi}{5} \end{bmatrix}, \quad k = 1, 2, \ldots, 5.$$

α értékére itt is az $\frac{1}{2} < \alpha \leqq \frac{\sqrt{5}-1}{2}$ megszorítást kell tennünk. $\frac{1}{2} < \alpha < \frac{\sqrt{5}-1}{2}$ esetén a **Z** mátrix reguláris, ezért a $\mathbf{v}_k$ vektorok lineárisan függetlenek. $\alpha = \frac{\sqrt{5}-1}{2}$ esetén viszont mindegyik $\mathbf{v}_k$ vektor két utolsó komponense 0, az ezek elhagyásával keletkező 3 dimenziós vektorok pedig az R^3 tér olyan P_1, P_2, P_3, P_4, P_5 pontjaihoz tartoznak, melyekre a $P_1P_2P_3P_4P_5$ ötszög egy síkbeli szabályos ötszög, s ennek középpontját az R^3 tér O kezdőpontjával összekötő egyenes merőleges az ötszög síkjára, a P_1OP_3 szög pedig derékszög.

7. Kopozitív mátrixok és kvadratikus programozás

A következőkben felsorakoztatunk néhány állítást, amelyek rávilágítanak a kopozitív mátrixok osztályának fontos szerepére a kvadratikus programozás elméletében.

11. TÉTEL. Tekintsük a

$$K = \{\mathbf{Qy}+\mathbf{Sz}: \mathbf{y} \geqq \mathbf{0}, \mathbf{z} \geqq \mathbf{0}, \mathbf{1}'\mathbf{z} = 1\}$$

konvex poliédert, ahol **Q** tetszőleges $n \times m_1$ mátrix $(m_1 \geqq 0)$, **S** tetszőleges $n \times m_2$ mátrix $(m_2 \geqq 1)$, Azt állítjuk, hogy

$$\mathbf{x}'\mathbf{Ax} + 2\mathbf{c}'\mathbf{x} - d \geqq 0 \tag{4}$$

akkor és csak akkor teljesül minden $\mathbf{x}\in K$ esetén, ha az $(m_1+m_2)\times(m_1+m_2)$ méretű

$$\overline{\mathbf{A}}-d\overline{\mathbf{B}}=\begin{bmatrix}\mathbf{Q}'\mathbf{A}\mathbf{Q} & \mathbf{Q}'\mathbf{A}\mathbf{S}+\mathbf{Q}'\mathbf{c}\mathbf{1}' \\ \mathbf{S}'\mathbf{A}\mathbf{Q}+\mathbf{1}\mathbf{c}'\mathbf{Q} & \mathbf{S}'\mathbf{A}\mathbf{S}+\mathbf{1}\mathbf{c}'\mathbf{S}+\mathbf{S}'\mathbf{c}\mathbf{1}'-d\mathbf{1}\mathbf{1}'\end{bmatrix}$$

mátrix kopozitív.

Itt $\mathbf{A}$ egy szimmetrikus $n\times n$ mátrix, $\mathbf{c}$ egy n dimenziós vektor,

$$\overline{\mathbf{A}}=\begin{bmatrix}\mathbf{Q}'\mathbf{A}\mathbf{Q} & \mathbf{Q}'\mathbf{A}\mathbf{S}+\mathbf{Q}'\mathbf{c}\mathbf{1}' \\ \mathbf{S}'\mathbf{A}\mathbf{Q}+\mathbf{1}\mathbf{c}'\mathbf{Q} & \mathbf{S}'\mathbf{A}\mathbf{S}+\mathbf{1}\mathbf{c}'\mathbf{S}+\mathbf{S}'\mathbf{c}\mathbf{1}'\end{bmatrix}$$

és

$$\overline{\mathbf{B}}=\begin{bmatrix}\mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}\mathbf{1}'\end{bmatrix}.$$

Bizonyítás. Tetszőleges $\mathbf{x}\in K$ esetén

$$\begin{aligned}&\mathbf{x}'\mathbf{A}\mathbf{x}+2\mathbf{c}'\mathbf{x}-d=\\ &=(\mathbf{y}'\mathbf{Q}'+\mathbf{z}'\mathbf{S}')\mathbf{A}(\mathbf{Q}\mathbf{y}+\mathbf{S}\mathbf{z})+2\mathbf{c}'(\mathbf{Q}\mathbf{y}+\mathbf{S}\mathbf{z})-d=\\ &=(\mathbf{y}'\mathbf{Q}'+\mathbf{z}'\mathbf{S}')\mathbf{A}(\mathbf{Q}\mathbf{y}+\mathbf{S}\mathbf{z})+2\mathbf{z}'\mathbf{1}\mathbf{c}'(\mathbf{Q}\mathbf{y}+\mathbf{S}\mathbf{z})-\\ &-d\mathbf{z}'\mathbf{1}\mathbf{1}'\mathbf{z}=[\mathbf{y}'\mathbf{z}'](\overline{\mathbf{A}}-d\overline{\mathbf{B}})\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix},\end{aligned}\tag{5}$$

ahol $\mathbf{y}\geqq\mathbf{0}$, $\mathbf{z}\geqq\mathbf{0}$ és $\mathbf{1}'\mathbf{z}=1$. Eszerint (4) akkor és csak akkor teljesül minden $\mathbf{x}\in K$ esetén, ha

$$\mathbf{v}'(\overline{\mathbf{A}}-d\overline{\mathbf{B}})\mathbf{v}\geqq 0\tag{6}$$

teljesül a

$$K_1=\left\{\mathbf{v}\colon \mathbf{v}=\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix},\ \mathbf{y}\in R_+^{m_1},\ \mathbf{z}\in R_+^{m_2},\ \mathbf{1}'\mathbf{z}=1\right\}$$

konvex poliéder vektoraira. $K_1\subset R_+^{m_1+m_2}$ miatt az utóbbi nyilvánvalóan következik az $\overline{\mathbf{A}}-d\overline{\mathbf{B}}$ mátrix kopozitivitásából. Ha viszont azt tudjuk, hogy (6) teljesül minden $\mathbf{v}\in K_1$ esetén, akkor minden olyan

$$\mathbf{v}=\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix}$$

vektorra is, melyre $\mathbf{y}\geqq\mathbf{0}$, $\mathbf{z}>\mathbf{0}$, fennáll

$$[\mathbf{y}'\ \mathbf{z}'](\overline{\mathbf{A}}-d\overline{\mathbf{B}})\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix}=(\mathbf{1}'\mathbf{z})^2\left[\frac{1}{\mathbf{1}'\mathbf{z}}\mathbf{y}'\ \ \frac{1}{\mathbf{1}'\mathbf{z}}\mathbf{z}'\right](\overline{\mathbf{A}}-d\overline{\mathbf{B}})\begin{bmatrix}\frac{1}{\mathbf{1}'\mathbf{z}}\mathbf{y}\\ \frac{1}{\mathbf{1}'\mathbf{z}}\mathbf{z}\end{bmatrix}\geqq 0.$$

Ebből pedig határátmenettel adódik, hogy

$$[\mathbf{y}'\ \mathbf{z}'](\overline{\mathbf{A}}-d\overline{\mathbf{B}})\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix}\geqq 0$$

teljesül akkor is, ha $\mathbf{y}\geqq\mathbf{0}$ és $\mathbf{z}=\mathbf{0}$.

12. TÉTEL. Legyenek K, $\mathbf{Q}$, $\mathbf{S}$, $\mathbf{A}$, $\mathbf{c}$, d, $\overline{\mathbf{A}}$ és $\overline{\mathbf{B}}$ ugyanazok, mint az előző tételnél, legyen továbbá

$$\mu_1 = \inf \{\mathbf{x}'\mathbf{A}\mathbf{x}+2\mathbf{c}'\mathbf{x}: \mathbf{x}\in K\}$$

és

$$\mu_2 = \sup \{d: \overline{\mathbf{A}}-d\overline{\mathbf{B}} \text{ kopozitív}\}.$$

Ekkor igazak a következők:

a) $\mu_1 = \mu_2$.

b) Index nélküli μ betűvel jelölve μ_1 és μ_2 közös értékét, véges μ esetén létezik olyan

$$\mathbf{v}_0\in R^{m_1+m_2}$$

vektor, melyre

$$\mathbf{v}_0 = \begin{bmatrix}\mathbf{y}_0\\ \mathbf{z}_0\end{bmatrix}, \quad \mathbf{y}_0\in R_+^{m_1}, \quad \mathbf{z}_0\in R_+^{m_2}, \quad \mathbf{z}_0 \neq 0, \tag{7}$$

és

$$\mathbf{v}_0'(\overline{\mathbf{A}}-\mu\overline{\mathbf{B}})\mathbf{v}_0 = 0. \tag{8}$$

Minden ilyen $\mathbf{v}_0$ vektorra

$$\mathbf{x}_0 = \frac{1}{\mathbf{1}'\mathbf{z}_0}(\mathbf{Q}\mathbf{y}_0+\mathbf{S}\mathbf{z}_0) \tag{9}$$

optimális megoldása a

$$\min \{\mathbf{x}'\mathbf{A}\mathbf{x}+2\mathbf{c}'\mathbf{x}: \mathbf{x}\in K\}$$

kvadratikus programozási feladatnak.

Bizonyítás. A 11. tétel szerint $\overline{\mathbf{A}}-d\overline{\mathbf{B}}$ akkor és csak akkor kopozitív, ha (4) teljesül minden $\mathbf{x}\in K$ esetén, vagyis ha $d\leqq\mu_1$. Ez éppen azt jelenti, hogy

$$\sup \{d: \overline{\mathbf{A}}-d\overline{\mathbf{B}} \text{ kopozitív}\} = \mu_1.$$

Ha μ véges, akkor a kvadratikus programozás alaptétele szerint van olyan $\mathbf{x}_0\in K$ vektor, melyre

$$\mathbf{x}_0'\mathbf{A}\mathbf{x}_0+2\mathbf{c}'\mathbf{x}_0 = \mu.$$

Mivel $\mathbf{x}_0=\mathbf{Q}\mathbf{y}_0+\mathbf{S}\mathbf{z}_0$ alakban írható, ahol $\mathbf{y}_0\in R_+^{m_1}$, $\mathbf{z}_0\in R_+^{m_2}$ és $\mathbf{1}'\mathbf{z}_0=1$, ezért az (5)-höz hasonló átalakítással adódik, hogy

$$0 = \mathbf{x}_0'\mathbf{A}\mathbf{x}_0+2\mathbf{c}'\mathbf{x}_0-\mu = [\mathbf{y}_0' \ \mathbf{z}_0'](\overline{\mathbf{A}}-\mu\overline{\mathbf{B}})\begin{bmatrix}\mathbf{y}_0\\ \mathbf{z}_0\end{bmatrix},$$

tehát $\mathbf{v}_0$ gyanánt választható az

$$\begin{bmatrix}\mathbf{y}_0\\ \mathbf{z}_0\end{bmatrix}$$

vektor.

Ha $\mathbf{v}_0$ egy tetszőleges, a (7) és (8) előírásokat kielégítő vektor, akkor a (9) szerint meghatározott $\mathbf{x}_0$ vektorra $\mathbf{x}_0\in K$ és

$$\mathbf{x}_0'\mathbf{A}\mathbf{x}_0+2\mathbf{c}'\mathbf{x}_0-\mu = \frac{1}{(\mathbf{1}'\mathbf{z}_0)^2}\mathbf{v}_0'(\overline{\mathbf{A}}-\mu\overline{\mathbf{B}})\mathbf{v}_0 = 0.$$

13. TÉTEL. Ha egy $q(\mathbf{x})=\mathbf{x}'\mathbf{A}\mathbf{x}+2\mathbf{c}'\mathbf{x}$ kvadratikus függvény korlátos a

$$K = K^{<}+K^{\triangle}$$

konvex poliéderen, ahol $K^{<}$ egy konvex poliedrikus kúp, $K^{\triangle}$ pedig egy nem üres korlátos konvex poliéder, akkor

a) $\mathbf{x}'\mathbf{A}\mathbf{x}=0$ minden $\mathbf{x}\in K^{<}$ esetén,

b) $\mathbf{x}_1'\mathbf{A}\mathbf{x}_2+\mathbf{x}_1'\mathbf{c}=0$, minden $\mathbf{x}_1\in K^{<}$, $\mathbf{x}_2\in K^{\triangle}$ esetén,

c) $q(\mathbf{x}_1+\mathbf{x}_2)=q(\mathbf{x}_2)$, minden $\mathbf{x}_1\in K^{<}$, $\mathbf{x}_2\in K^{\triangle}$ esetén.

Bizonyítás. Tegyük fel, hogy

$$d_1 \leqq q(\mathbf{x}) \leqq d_2$$

a K halmazon, és legyen

$$K^{<} = \{\mathbf{Q}\mathbf{y}\colon \mathbf{y} \geqq \mathbf{0}\},$$
$$K^{\triangle} = \{\mathbf{S}\mathbf{z}\colon \mathbf{z} \geqq \mathbf{0},\ \mathbf{1}'\mathbf{z} = 1\},$$

ahol $\mathbf{Q}$ egy $n\times m_1$ méretű mátrix, $\mathbf{S}$ pedig egy $n\times m_2$ méretű mátrix ($m_2\geqq 1$). A 11. tétel alapján adódik, hogy a $\mathbf{Q}'\mathbf{A}\mathbf{Q}$ és a $-\mathbf{Q}'\mathbf{A}\mathbf{Q}$ mátrix egyaránt kopozitív. Ez csak úgy lehetséges, hogyha

$$\mathbf{Q}'\mathbf{A}\mathbf{Q} = \mathbf{0}, \tag{10}$$

ekkor pedig az 1. tétel segítségével adódik, hogy

$$\mathbf{Q}'\mathbf{A}\mathbf{S}+\mathbf{Q}'\mathbf{c}\mathbf{1}' = \mathbf{0}. \tag{11}$$

Tegyük fel, hogy $\mathbf{x}\in K^{<}$. Ekkor $\mathbf{x}=\mathbf{Q}\mathbf{y}$, ahol $\mathbf{y}\geqq\mathbf{0}$, így (10) miatt

$$\mathbf{x}'\mathbf{A}\mathbf{x} = \mathbf{y}'\mathbf{Q}'\mathbf{A}\mathbf{Q}\mathbf{y} = 0.$$

Most tegyük fel, hogy $\mathbf{x}_1\in K^{<}$ és $\mathbf{x}_2\in K^{\triangle}$. Ekkor $\mathbf{x}_1=\mathbf{Q}\mathbf{y}$, ahol $\mathbf{y}\geqq\mathbf{0}$ és $\mathbf{x}_2=\mathbf{S}\mathbf{z}$, ahol $\mathbf{z}\geqq\mathbf{0}$, $\mathbf{1}'\mathbf{z}=1$, így (10) és (11) miatt

$$\mathbf{x}_1'\mathbf{A}\mathbf{x}_2+\mathbf{x}_1'\mathbf{c} = \mathbf{y}'\mathbf{Q}'\mathbf{A}\mathbf{S}\mathbf{z}+\mathbf{y}'\mathbf{Q}'\mathbf{c}\mathbf{1}'\mathbf{z} =$$
$$= \mathbf{y}'(\mathbf{Q}'\mathbf{A}\mathbf{S}+\mathbf{Q}'\mathbf{c}\mathbf{1}')\mathbf{z} = 0.$$

Végül a 13. tétel már bebizonyított a) és b) állításából azonnal adódik, hogy

$$q(\mathbf{x}_1+\mathbf{x}_2)-q(\mathbf{x}_2) = \mathbf{x}_1'\mathbf{A}\mathbf{x}_1+2(\mathbf{x}_1'\mathbf{A}\mathbf{x}_2+\mathbf{x}_1'\mathbf{c}) = 0.$$

Megjegyzések és problémák felvetése

A *Motzkin-féle felbontási tétel* szerint bármely K konvex poliéder megadható a 11. tételben szereplő módon. A $\mathbf{Q}$ mátrix oszlopait alkotó vektorok gyanánt választhatók a K poliéder extremális irányai. Ha K nem tartalmaz egyenest, akkor az $\mathbf{S}$ mátrix oszlopait alkotó vektorok gyanánt választhatók a K poliéder extremális pontjai. Ily módon az $m_2\geqq 1$ feltevés elég természetesnek tekinthető. Ha K konvex poliedrikus kúp és

$$K = \{\mathbf{Q}\mathbf{y}\colon \mathbf{y} \geqq \mathbf{0}\},$$

akkor a hiányzó $\mathbf{S}$ mátrix minden esetben pótolható egy $\mathbf{0}$ oszlopvektorral.

A 12. tételnek a gyakorlatban való alkalmazhatóságát már az nagyon korlátozza, hogy ismernünk kell a K poliéder extremális irányait és extremális pontjait. (Ezek meghatározására szolgáló módszerek találhatók az [1], [14] és [17] irodalmi hivatkozásokban.) Mégsem tartom teljesen reménytelennek, hogy a 12. tétel továbbfejlesztése által lehetséges legyen a gyakorlatban is alkalmazható eljárást találni az általános kvadratikus programozási feladat megoldására.

A 12. tétel lényegének az általános (azaz nem okvetlenül konvex) kvadratikus programozási feladat és egy, kopozitív kvadratikus alakok segítségével megfogalmazott, feladat ekvivalenciájának a felismerését tartom. Ennek alapján az általános kvadratikus programozási feladat megoldása esetleg megvalósítható a következő két részfeladat megoldása útján:

a) $\mu=\sup\{d:\overline{\mathbf{A}}-d\overline{\mathbf{B}}$ kopozitív$\}$ értékének a meghatározása,

b) legalább egy olyan

$$\mathbf{v}_0=\begin{bmatrix}\mathbf{y}_0\\ \mathbf{z}_0\end{bmatrix}$$

vektor meghatározása, melyre $\mathbf{y}_0\in R_+^{m_1}$, $\mathbf{z}_0\in R_+^{m_2}$, $\mathbf{z}_0\neq\mathbf{0}$ és $\mathbf{v}_0'(\overline{\mathbf{A}}-\mu\overline{\mathbf{B}})\mathbf{v}_0=0$.

A tárgyhoz kapcsolódó problémaként merül fel hatékony algoritmusok kidolgozásának az igénye az a) és b) feladatok numerikus megoldására.

Egy másik, elvi és gyakorlati szempontból egyaránt fontos kérdés lenne annak megvizsgálása, hogy nem lehet-e a 12. tétel állítását olyan módon továbbfejleszteni, hogy megszabaduljunk a K poliéder extremális irányai és extremális pontjai explicit ismeretének a szükségességétől.

IRODALOM

[1] BALINSKI, M. L., "An algorithm for finding all vertices of convex polyhedral sets", *J. Soc. Indust. Appl. Math.* **9** (1961) 72—88.

[2] BASTON, V., "Extreme copositive quadratic forms", *Acta Arith.* **15** (1969) 319—327.

[3] BAUMERT, L. D., "Extreme copositive quadratic forms", *Pacific J. Math.* **19** (1966) 197—204.

[4] BAUMERT, L. D., "Extreme copositive quadratic forms II", *Pacific J. Math.* **20** (1967) 1—20.

[5] BLUM, E. and OETTLI, W., "Direct proof of the existence theorem for quadratic programming", *Operations Research* **20** (1972) 165—167.

[6] COTTLE, R. W. and DANTZIG, G. B., "Complementary pivot theory of mathematical programming", *Linear Algebra and Appl.* **1** (1968) 103—125.

[7] COTTLE, R. W., HABETLER, G. J. and LEMKE, C. E., "On classes of copositive matrices", *Linear Algebra and Appl.* **3** (1970) 295—310.

[8] DIANANDA, P. H., "On non-negative forms in real variables some or all of which are non-negative", *Proc. Cambridge Phil. Soc.* **58** (1962) 17—25.

[9] FRANK, M. and WOLFE, P., "An algorithm for quadratic programming", *Nav. Res. Log. Qu.* **3** (1956) 95—110.

[10] HALL, M., "Discrete problems", in: *A Survey of Numerical Analysis* Ed. J. Todd (New York, 1962) 518—542.

[11] HALL, M. and NEWMAN, M., "Copositive and completely positive quadratic forms", *Proc. Cambridge Phil. Soc.* **59** (1963) 329—339.

[12] HAYNSWORTH, E. and HOFFMAN, A. J., "Two remarks on copositive matrices", *Linear Algebra and Appl.* **2** (1969) 387—392.

[13] HOFFMAN, A. J. and PEREIRA, F., "On copositive matrices with $-1, 0, 1$ entries", *J. Comb. Theory (A)* **14** (1973) 302—309.

[14] JAHANSHAHLOU, G. R. and MITRA, G., "Two algorithms for finding all the vertices of a convex polyhedron", in: *Progress in Operation Research* Ed. A. Prékopa (North-Holland, 1976) 503—535.

[15] KÉRI, G., "On the maximum value of a quadratic function under linear constraints", *Studia, Sci. Math. Hung.* **6** (1971) 193—196.
[16] KÉRI, G., "An examination of nonnegativity and quasiconvexity conditions of quadratic forms on the nonnegative orthant", *Studia Sci. Math. Hung.* **7** (1972) 11—20.
[17] MANAS, M. and NEDOMA, J., "Finding all the vertices of a convex polyhedron", *Numerische Mathematik* **12** (1968) 226—229.
[18] MOTZKIN, T. S., *Beiträge zur Theorie der linearen Ungleichungen* (University of Basel, Jerusalem 1936).
[19] MOTZKIN, T. S. and STRAUS, E. G., "Maxima for graphs and a new proof of a theorem of Turán", *Canad. J. Math.* **17** (1965) 533—540.

(Beérkezett: 1976. október 15.)

KÉRI GERZSON
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST I., ÚRI U. 49.
NEMZETKÖZI TUDÓSKOLLEKTÍVA, MOSZKVA

ON A CLASS OF QUADRATIC FORMS

G. KÉRI

In the first part (Section 1—6.) of the paper some theorems for the characterization of copositive matrices and some counterexamples about such matrices are given. Some of the results presented here — as the author has been acquainted lately — were known already. (See references [2]—[4], [8] and [10]—[13].)

In the second part (Section 7.) it is pointed out that the general (i.e. not necessarily convex) quadratic programming problem can be expressed as an equivalent problem concerning copositive matrices. On the score of this, the framework of a potential method of solution of the general quadratic programming problem can be formed. After indicating the difficulties connected with the outlined method, some problems directed to the elimination of these difficulties are posed.

Alkalmazott Matematikai Lapok **2** *(1976) 413—425*

A LAGRANGE SZORZÓK HASZNÁLATA DISZKRÉT PROGRAMOZÁSI ALGORITMUSOKBAN

VIZVÁRI BÉLA
Budapest

A dolgozat célja, hogy megvizsgáljuk, miként hasznosíthatók a *Lagrange szorzók* egy komplex diszkrét programozási algoritmusban. Először egy új kritériumot adunk meg, amelynek teljesülése garantálja egy *Lagrange szorzók* által generált pont optimalitását. Ezután megmutatjuk, hogy *Lagrange szorzók* segítségével hogyan lehet lerövidíteni a leszámlálási algoritmusok végrehajtásához szükséges számításokat. Végül a *Lagrange szorzók* és az ún. *s-feltétel* kapcsolatát vizsgáljuk és ebből vonunk le algoritmikus következtetéseket.

1. Bevezetés

Azok a módszerek, amelyekkel ebben a cikkben foglalkozni fogunk, H. EVERETT [3] alapvető eredményeire támaszkodnak. Az eljárásnak diszkrét programozásban számos alkalmazása ismeretes ([7], [10], [11], [12], [13], [14], [15]). Az eredeti módszer esetén felmerülnek bizonyos nehézségek, amelyeket az eljárással együtt ismertetni fogunk. A dolgozat egyik célja ezekenek a problémáknak a kiküszöbölése.

A következő feladattal foglalkozunk:

$$\max f(\mathbf{x})$$

$$\mathbf{g}(\mathbf{x}) \leqq \mathbf{b} \tag{1.1}$$

$$\mathbf{x} \in S,$$

ahol $\mathbf{x} \in \mathbf{R}^n$; $S \subset \mathbf{R}^n$; $\mathbf{g}(\mathbf{x}), \mathbf{b} \in \mathbf{R}^m$; $f(\mathbf{x})$ és a $\mathbf{g}(\mathbf{x})$ függvény komponensei tetszőleges függvények.

A feladat ilyen alakban való megfogalmazásának az az értelme, hogy a feltételek két részre oszthatók. Az első részben az analitikusan jól megfogalmazható követelmények találhatók, míg a második részben a logikai feltételek, vagy egyéb analitikusan nehezen kezelhető megszorítások állnak. Tipikus esete ennek a nulla-egyes diszkrét programozási feladat:

$$\max \mathbf{c}'\mathbf{x}$$

$$\mathbf{A}\mathbf{x} \leqq \mathbf{b} \tag{1.2}$$

$$\mathbf{x} \in \mathbf{D}^n$$

ahol $\mathbf{x}, \mathbf{c} \in \mathbf{R}^n$; $\mathbf{b} \in \mathbf{R}^m$.

$\mathbf{A}$ $m \times n$-es mátrix; és

$$\mathbf{D}^n = \{\mathbf{y}: \mathbf{y} \in \mathbf{R}^n \cdot y_j = 0 \text{ vagy } 1,\ j = 1, \ldots, n\}.$$

Egy nemnegatív $\lambda \in R^m$ vektor segítségével kaphatjuk az ún. Lagrange-feladatot:

$$\max_{x \in S} [f(\mathbf{x}) - \lambda' \mathbf{g}(\mathbf{x})] \tag{1.3}$$

H. EVERETT már említett munkájában bizonyította, hogy igaz az

1.1. TÉTEL: Ha egy adott λ esetén (1.3) optimális megoldása $\mathbf{x}^*$, akkor $\mathbf{x}^*$ optimális megoldása (1.1)-nek is, ha $\mathbf{b} = \mathbf{g}(\mathbf{x}^*)$.

A bizonyítást az olvasóra bízzuk. Ennek a tételnek az átfogalmazásával R. BROOKS és A. GEOFFRION [2] egy optimalitási kritériumot talált:

1. *Optimalitási kritérium:* Ha $\mathbf{x}^*$ (1.3) optimális megoldása egy adott λ-ra, akkor ha

$$\lambda_i > 0 \quad \text{esetén} \quad g_i(\mathbf{x}^*) = b_i$$

és

$$\lambda_i = 0 \quad \text{esetén} \quad g_i(\mathbf{x}^*) \leqq b_i,$$

úgy $\mathbf{x}^*$ optimális megoldása az (1.1) feladatnak is.

Most bevezetünk két fogalmat, amelyekre a későbbiek során szükségünk lesz.

1.1. DEFINÍCIÓ: A λ vektor az $\mathbf{x}^*$ pontot generálja, ha az (1.3) feladatnak $\mathbf{x}^*$ optimális megoldása.

1.2. DEFINÍCIÓ: Az $\mathbf{x}^*$ pont generálható, ha létezik olyan λ vektor, hogy a megfelelő (1.3) feladatnak $\mathbf{x}^*$ optimális megoldása.

Az (1.3) probléma általában jóval egyszerűbb mint az eredeti (1.1) feladat. Ez különösen igaz a diszkrét programozás esetében, a *Lagrange-probléma* itt ugyanis a következő:

$$\max_{x \in D^n} [\mathbf{c}'\mathbf{x} - \lambda'\mathbf{A}\mathbf{x}] = \max_{x \in D^n} [\mathbf{c}' - \lambda'\mathbf{A}]\mathbf{x}. \tag{1.4}$$

Tehát (1.4) megoldásához csak a $\mathbf{c}' - \lambda'\mathbf{A}$ vektor komponenseinek az előjelét kell megvizsgálni. Ez adja a gondolatot, hogy az eredeti (1.1) feladat helyett *Lagrange-problémák* egy sorozatát oldjuk meg. Egy ilyen algoritmus k. lépése a következő:

(k1) A korábbi λ_l szorzók és az általuk generált $\mathbf{x}_l^*$ pontok ismeretében $(l = 1, \ldots, k-1)$ megválasztjuk a λ_k vektort.

(k2) A *Lagrange-problémá*ból meghatározzuk az $\mathbf{x}_k^*$ pontot.

(k3) Ha teljesül az optimalitási kritériumunk, akkor az eljárás véget ért. Ellenkező esetben $k = k+1$.

Egy ilyen algoritmus kapcsán több nehézség is felmerül. Az elsőt már maga EVERETT is felismerte. Arról van szó, hogy az (1.1) feladat megengedett tartományának nem minden pontja generálható. A nem generálható pontok közé eshet véletlenül az optimális megoldás is. EVERETT adott meg módszereket ilyen pontok kezelésére. Rajta kívül még mások is megvizsgálták ezt a kérdést ([2], [11]). Diszkrét programozásban, mint azt később majd megmutatjuk, a probléma végső soron elkerülhető.

A második nehézség lényege az, hogy az *1. Optimalitási kritérium* csak elegendő feltétele az optimalitásnak, de nem szükséges feltétel is. Tekintsük ugyanis a következő feladatot:

$$\begin{aligned} &\max 3x_1+2x_2+6x_3 \\ &x_1+\ x_2+2x_3 \leqq 3 \\ &2x_1-\ x_2+3x_3 \leqq 4 \\ &x_1, x_2, x_3 = 0, \text{ vagy } 1. \end{aligned} \tag{1.5}$$

Könnyen látható, hogy (1.5) optimális megoldása:

$$x_1^* = 0, \quad x_2^* = 1, \quad x_3^* = 1.$$

A megfelelő jobboldalak: 3, ill. 2. $\mathbf{x}^*$ generálható, például a $\lambda^{*\prime}=(1, 1;\ 1, 1)$ vektorral. Ha azonban az *1. Optimalitási kritérium*ot akarjuk alkalmazni, akkor a jobboldalak miatt

$$\lambda' = (a, 0); \quad a > 0$$

alakú szorzókra van szükségünk. Ilyen λ vektor azonban sohasem fogja $\mathbf{x}^*$-ot generálni, és így az *1. Optimalitási kritérium* nem fog működni ebben az esetben.

2. Egy új optimalitási kritérium

Tegyük fel, hogy λ^* generálja $\mathbf{x}^*$-ot, amely az (1.1) feladat megengedett megoldása. Ekkor tudjuk, hogy egy tetszőleges $\mathbf{x}\in S$ esetén

$$f(\mathbf{x}^*)-\lambda^{*\prime}\mathbf{g}(\mathbf{x}^*) \geqq f(\mathbf{x})-\lambda^{*\prime}\mathbf{g}(\mathbf{x}),$$

amiből

$$f(\mathbf{x}^*)-f(\mathbf{x}) \geqq \lambda^{*\prime}[\mathbf{g}(\mathbf{x}^*)-\mathbf{g}(\mathbf{x})]. \tag{2.1}$$

Látható, hogy ahol (2.1) jobboldala nemnegatív, ott $\mathbf{x}^*$ nemcsak megengedett, hanem optimális megoldás is.

2.1. DEFINÍCIÓ: Az $\mathbf{x}^*$ pont optimalitási tartományának nevezzük a

$$H(\mathbf{x}^*, \lambda^*) = \{\mathbf{x}:\ \mathbf{x}\in S,\ \lambda^{*\prime}[\mathbf{g}(\mathbf{x}^*)-\mathbf{g}(\mathbf{x})] \geqq 0\}$$

halmazt, $\mathbf{x}^*$ az (1.1) feladat megengedett megoldása, melyet a λ^* pont generál.

Most már megfogalmazható a

2. Optimalitási kritérium: Tegyük fel, hogy az eljárás során a $\lambda_1, \ldots, \lambda_r$ vektorokkal az (1.1) probléma $\mathbf{x}_1, \ldots, \mathbf{x}_r$ megengedett megoldásait generáltuk. Legyen

$$H = \bigcup_{i=1}^{r} H(\mathbf{x}_i, \lambda_i)$$

és

$$P = \{\mathbf{x}:\ \mathbf{x}\in S,\ \mathbf{g}(\mathbf{x}) \leqq \mathbf{b}\},$$

továbbá

$$\max_{1\leqq i\leqq r} f(\mathbf{x}_i) = f(\mathbf{x}_j).$$

Ha a

$$P \subset H \tag{2.2}$$

reláció teljesül, akkor $\mathbf{x}_j$ optimális megoldása az (1.1) feladatnak.

Röviden összefoglalva: az eddig generált legjobb megengedett megoldás optimális lesz, ha a generált megengedett megoldások optimalitási tartományai lefedik az (1.1) probléma megengedett tartományát.

A (2.2) összefüggést természetesen számítástechnikailag könnyen kezelhető alakra kell hozni. Ezt a diszkrét programozási esetben tesszük meg.

Célszerűbb (2.2)-t a vele ekvivalens

$$P \cap \bar{H} = \emptyset \tag{2.3}$$

formában használni, ahol $\bar{H}$ a H komplementere, $\emptyset$ pedig az üres halmazt jelöli. H definíciója alapján igaz a

$$\bar{H} = \bigcap_{i=1}^{r} \bar{H}(\mathbf{x}_i, \lambda_i)$$

összefüggés. A diszkrét programozási esetben

$$\bar{H}(\mathbf{x}_i, \lambda_i) = \{\mathbf{x}\colon \mathbf{x} \in \mathbf{D}^n;\ \lambda_i \mathbf{A}\mathbf{x}_i - \lambda_i \mathbf{A}\mathbf{x} < 0\}.$$

Vegyük észre, hogy ekkor (2.3) az $\mathbf{x} \in D^n$ követelménytől eltekintve csak lineáris feltételekből áll. Ez adja a gondolatot a következő eljáráshoz. Legyen

$$\mathbf{K}^n = \{\mathbf{x}\colon \mathbf{x} \in \mathbf{R}^n;\ 0 \leqq x_j \leqq 1,\ j = 1, \ldots, n\},$$

továbbá

$$G(\mathbf{x}_i, \lambda_i) = \{\mathbf{x};\ \mathbf{x} \in \mathbf{K}^n;\ \lambda_i \mathbf{A}\mathbf{x}_i - \lambda_i \mathbf{A}\mathbf{x} \leqq -\varepsilon\}, \tag{2.4}$$

ahol ε egy kicsiny nemnegatív szám. $\varepsilon = 0$ esetén nyilván

$$\bar{H}(\mathbf{x}_i, \lambda_i) \subset G(\mathbf{x}_i, \lambda_i)$$

reláció teljesül. (2.3) baloldalán diszkrét pontok egy halmaza áll. Ezt a halmazt helyettesítjük a

$$Q = \{\mathbf{x}\colon \mathbf{x} \in \mathbf{K}^n;\ \mathbf{A}\mathbf{x} \leqq \mathbf{b};\ \mathbf{x} \in G(\mathbf{x}_i, \lambda_i),\ i = 1, \ldots, r\}$$

konvex poliéderrel. Mi a

$$Q = \emptyset$$

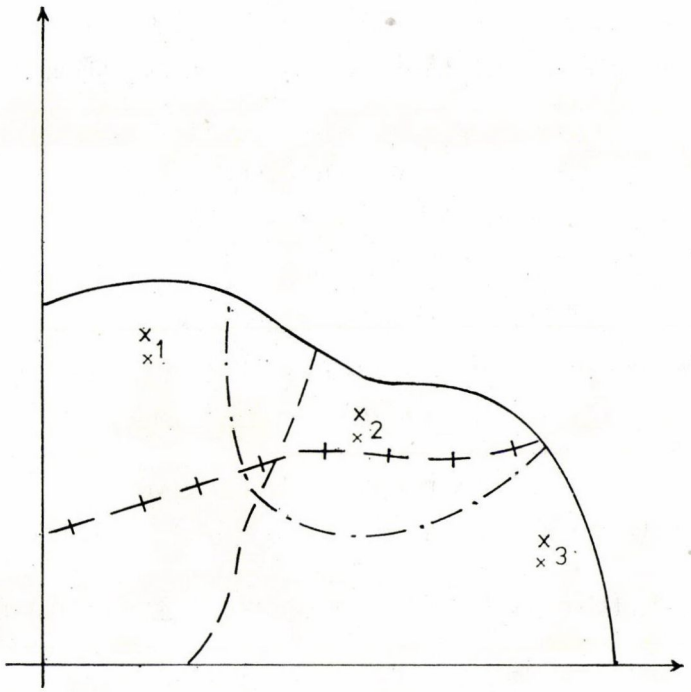
reláció teljesülését fogjuk vizsgálni, megoldva a

$$\max_{\mathbf{x} \in Q} \mathbf{c}'\mathbf{x} \tag{2.5}$$

lineáris programozási feladatot. Legyen az optimális célfüggvényérték z_0. Ha

$$z_0 \leqq \max_{i=1,\ldots,r} \mathbf{c}'\mathbf{x}_i \tag{2.6}$$

akkor úgy tekintjük, hogy (1.2)-nek nincs olyan megengedett megoldása, mely jobb célfüggvényértékkel rendelkezik, mint az eddig talált legjobb megengedett megoldás. ($Q = \emptyset$ esetén $z_0 = -\infty$.) Természetesen (2.6) csak $\varepsilon = 0$ esetén bizonyító erejű.



1. ábra

Ekkor viszont az $\mathbf{x}_1, \ldots, \mathbf{x}_r$ pontokat nem zártuk ki, így (2.6)-ban csak egyenlőség teljesülhet.

Folytatva az 1. szakasz példáját, megmutatjuk, hogy a 2. Optimalitási kritérium bebizonyítja az $\mathbf{x}^{*\prime}=(0, 1, 1)$ pont optimalitását. $\lambda^{*\prime}=(1, 1; 1, 1)$ esetén ugyanis

$$\overline{H}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \{\mathbf{x}: \mathbf{x}\in \mathbf{D}^3;\ -3{,}3x_1 - 5{,}5x_3 < -5{,}5\}.$$

Innen

$$\overline{H}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \{(1, 0, 1),\ (1, 1, 1)\}.$$

Könnyen látható, hogy

$$P \cap \overline{H}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \emptyset,$$

ami a keresett állítást adja.

Igaz továbbá a

2.1. TÉTEL: Ha az (1.1) feladat $\mathbf{x}^*$ megengedett megoldását a λ^* vektor generálja és

$$\lambda_i^* > 0 \quad \text{esetén} \quad g_i(\mathbf{x}^*) = b_i$$

és

$$\lambda_i^* = 0 \quad \text{esetén} \quad g_i(\mathbf{x}^*) \leqq b_i$$

akkor

$$P \subset H(\mathbf{x}^*, \lambda^*), \tag{2.7}$$

ahol

$$P = \{\mathbf{x}: \mathbf{x} \in S;\ \mathbf{g}(\mathbf{x}) \leqq \mathbf{b}\}.$$

Bizonyítás. Mivel $\mathbf{x}^*$-ot λ^* generálja, ezért minden $\mathbf{x} \in S$ esetén

$$f(\mathbf{x}^*) - \lambda^{*\prime}\mathbf{g}(\mathbf{x}^*) \geqq f(\mathbf{x}) - \lambda^{*\prime}\mathbf{g}(\mathbf{x}).$$

Innen átrendezéssel

$$f(\mathbf{x}^*) - f(\mathbf{x}) \geqq \lambda^{*\prime}[\mathbf{g}(\mathbf{x}^*) - \mathbf{g}(\mathbf{x})]. \tag{2.8}$$

Tegyük fel, hogy $\mathbf{x}$ megengedett pont, vagyis $\mathbf{x} \in P$. Ekkor (2.8) jobboldala tovább alakítható:

$$\lambda^{*\prime}[\mathbf{g}(\mathbf{x}^*) - \mathbf{g}(\mathbf{x})] = \sum_{\lambda_i^* > 0} \lambda_i^* [g_i(\mathbf{x}^*) - g_i(\mathbf{x})] = \sum_{\lambda_i^* > 0} \lambda_i^* [b_i - g_i(\mathbf{x})] \geqq 0,$$

ami azt mutatja, hogy $\mathbf{x} \in H(\mathbf{x}^*, \lambda^*)$.

A (2.7) reláció azt jelenti, hogy valahányszor az *1. Optimalitási kritérium* bizonyítja egy pont optimalitását, ugyanezt kimutatja a *2. Optimalitási kritérium* is. Tehát, mint azt a számpélda igazolja, az utóbbi szélesebb körben alkalmazható.

3 Lagrange-szorzók leszámlálási algoritmusokban

Ismeretes, hogy a leszámlálási algoritmusok két fázisból állnak. Az első fázisban egyre nagyobb és nagyobb célfüggvényértékkel rendelkező megengedett megoldásokat keresünk (az (1.2) alakú feladatra gondolva!), a második fázisban pedig az addig talált legjobb megengedett megoldás optimalitását bizonyítjuk. Egy konkrét számítás során nem tudjuk, hogy mikor ért véget az első fázis, mivel az algoritmus mindkettőben ugyanazokat a lépéseket végzi. A második fázis bizonyos értelemben felesleges. Nélküle a megtalált optimális pontról csak annyit tudunk, hogy egy nagyon jó megengedett megoldás, míg a második fázis után tudjuk az optimalitását is, de a pont maga természetszerűleg nem változott meg. Vagyis nagyon kevés az a többlet információ, amit az eljárásnak ebből a részéből nyerhettünk. Ennek ellenére a második fázis a szükséges számítási időnek jóval több mint a felét, mintegy a háromnegyedét emészti fel. Felmerül tehát a kérdés, hogyan lehet a második fázis végrehajtását felgyorsítani. Ehhez az éppen vizsgált feladatról olyan új információkra van szükségünk, amelyeket a számítás során nyerhetünk.

Ilyenek lehetnek például

— a célfüggvény feltétel,

— a *Lagrange-szorzók* által generált (2.4) alakú feltételek.

Mielőtt ezeknek a használatára rátérnénk, egy rövid áttekintését adjuk a leszámlálási algoritmusoknak. Itt nincs mód és hely az egzakt matematikai tárgyalásra. A *pszeudo-megoldás-fa* fogalma BALASTÓL származik ([1]), míg a leszámlálási algoritmusok általános vázát KOVÁCS LÁSZLÓ BÉLA publikálta ([8], [9]).

Az (1.2) feladatra vonatkoztatva megoldás alatt mindig a $\mathbf{D}^n$ pontjait értjük, megengedett megoldás alatt pedig az összes feltételt kielégítő pontokat.

Az algoritmus során a változókat mindig két részre osztjuk. Az elsőbe tartozó változók értékeit már 0 vagy 1 szinten rögzítettük, míg a többi változó értéke még meghatározatlan, ez utóbbiakat nevezzük szabad változóknak. Az eljárás lényege, hogy az eddigi rögzítésekből következtetéseket vonjunk le. Ezek az alábbi két fő típusba tartoznak:

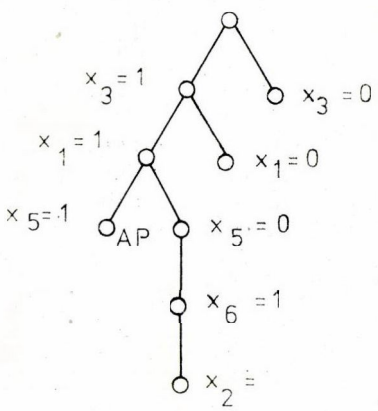
a) A jelenlegi rögzítések mellett nincs megengedett megoldás.

b) A jelenlegi rögzítések egyértelműen meghatározzák egyes szabad változók értékét. Ez azt jelenti, hogy ha a változó az ellenkező értéket venné fel, akkor nem lenne megengedett megoldás. Az ilyen következményekből adódó rögzítéseket lekötéseknek nevezzük.

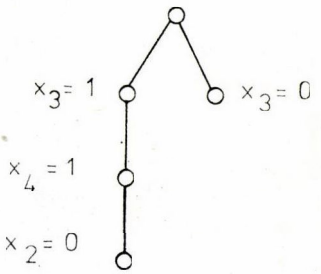
A pillanatnyilag érvényben levő rögzítések egy pszeudo-megoldást határoznak meg, ami nem más, mint az ezeknek a rögzítéseknek eleget tevő megengedett megoldások halmaza. Az algoritmus során a pszeudo-megoldások együttesen egy *pszeudomegoldás-fát* adnak, amit a rögzítések sorrendje határoz meg. Az eljárás folyamán ezen fa valamennyi ágát meg kell vizsgálnunk az algoritmus ágcsere (angolul *backtrack*) lépéseinek megfelelően.

Most egy példán keresztül mutatjuk meg, hogy a feladatról kapott új információk birtokában hogyan lehet a *pszeudomegoldás-fa* egyes ágait átugrani az eljárás folyamán.

A 2. ábrán egy *pszeudomegoldás-fát* látunk. A számítás folyamán először az x_3 kapta az 1 értéket. Ezt követte az $x_1=1$ rögzítés stb. Az $x_6=1$ és az $x_2=0$ lekötések a korábbi rögzítések következményei. Ha itt egy ágcserét kell végrehajtanunk, akkor az *AP*-vel jelölt pontot kapjuk. Azonban a következőt is megtehetjük. Visszatérünk a *pszeudomegoldás-fa* egy korábbi állapotához, például az $x_3=1$ rögzítésnek megfelelő részeket tartjuk meg. Ekkor az új feltételekből megpróbálunk lekötéseket nyerni. Ha ez sikerült, mint az a 3. ábrán látható esetben történt, akkor a kapott



2. ábra



3. ábra

pszeudomegoldás-fa lényegesen jobb annál, amelyikből kiindultunk (példánkban a 2. ábrán láthatónál). Amikor ilyen új következményeket akarunk találni, akkor csak az új feltételeket használhatjuk, hiszen ha az eredeti feltételekből le lehetne vonni következtetéseket, akkor ezt már korábban megtehettük volna, mikor eredetileg eljutottunk a *pszeudomegoldás-fa* most vizsgált állapotába.

Világos, hogy a fenti eljárással nem hagyhatunk ki olyan megengedett megoldást, mely jobb célfüggvényértékkel rendelkezik, mint az eddig talált legjobb megengedett megoldás, mert az előbbi pontok kielégítik az új feltételeket is.

A fentiek alapján a *Lagrange-szorzók* leszámlálási algoritmusokban való használatának az alapgondolata a következő. Amikor az eljárás során egyre jobb és jobb megengedett megoldásokat határozunk meg, akkor megkeressük ezekhez a pontokhoz azokat a Lagrange-szorzókat, amelyek őket generálják. Így (2.3) típusú feltételeket tudunk generálni, amelyek segítségével a *pszeudomegoldás-fa* egyes ágait át tudjuk ugorni.

Egy új kérdés merül tehát fel, hogyan lehet egy adott megengedett megoldáshoz őt generáló *Lagrange-szorzókat* találni.

Először a szorzók egzisztenciájának kérdését vizsgáljuk, mert az 1. szakasz alapján tudjuk, hogy ilyenek nem feltétlenül léteznek. Tehát tekintjük a

$$\begin{aligned} &\max \mathbf{c}'\mathbf{x} \\ &\mathbf{A}\mathbf{x} \leqq \mathbf{b} \\ &\mathbf{x} \in \mathbf{D}^n \end{aligned} \tag{3.1}$$

feladatokat. Itt az általánosság megszorítása nélkül feltehetjük, hogy a $\mathbf{c}$ vektor komponensei nemnegatívok. Adjuk hozzá a (3.1) feladathoz az

$$x_j \leqq 1, \quad j = 1, \ldots, n$$

„új" feltételeket. Ez természetesen nem jelent további megszorítást, hiszen $\mathbf{D}^n \subset \mathbf{K}^n$. Így feladatunk alakja a következő lesz:

$$\begin{aligned} &\max \mathbf{c}'\mathbf{x} \\ &\mathbf{A}\mathbf{x} \leqq \mathbf{b} \\ &\mathbf{E}\mathbf{x} \leqq \mathbf{e} \\ &\mathbf{x} \in \mathbf{D}^n \end{aligned}$$

ahol $\mathbf{E}$ az n dimenziós egységmátrix és az $\mathbf{e}$ vektor minden komponense 1. Az első két feltételnek megfelelő *Lagrange-szorzók* legyenek a $\boldsymbol{\lambda}$, illetve a $\boldsymbol{\mu}$ vektorok. Adott egy $\bar{\mathbf{x}}$ megoldás. Megadjuk a $\boldsymbol{\lambda}$ és $\boldsymbol{\mu}$ vektor egy olyan választását, hogy $\bar{\mathbf{x}}$ a megfelelő *Lagrange-feladat* optimális megoldása lesz. Legyen

$$\boldsymbol{\lambda} = \mathbf{0}$$

$$\mu_j = \begin{cases} 0, & \text{ha} \quad \bar{x}_j = 1 \\ c_j + 1, & \text{ha} \quad \bar{x}_j = 0. \end{cases} \tag{3.2}$$

A *Lagrange-feladat:*

$$\max_{\mathbf{x} \in \mathbf{D}^n} [\mathbf{c}'\mathbf{x} - \boldsymbol{\lambda}'\mathbf{A}\mathbf{x} - \boldsymbol{\mu}'\mathbf{E}\mathbf{x}] = \max_{\mathbf{x} \in \mathbf{D}^n} [\mathbf{c}' - \boldsymbol{\mu}']\mathbf{x}.$$

A célfüggvény komponensei:

$$c_j-\mu_j=\begin{cases} c_j, & \text{ha} \quad \bar{x}_j=1, \\ -1, & \text{ha} \quad \bar{x}_j=0, \end{cases}$$

ami $\bar{\mathbf{x}}$ optimalitását igazolja.

A (3.2) konstrukció természetesen csak az egzisztencia igazolására szolgál. A továbbiakban a szorzók jó megválasztásával foglalkozunk.

Tegyük fel, hogy a λ^* generálja $\mathbf{x}^*$-ot. Ez tehát azt jelenti, hogy

$$\mathbf{c}'\mathbf{x}^*-\lambda^{*\prime}\mathbf{A}\mathbf{x}^* \geqq \mathbf{c}'\mathbf{x}-\lambda^{*\prime}\mathbf{A}\mathbf{x}$$

minden $\mathbf{x}\in\mathbf{D}^n$ esetén. Innen átrendezéssel adódik a

$$\mathbf{c}'\mathbf{x}^*+\lambda^{*\prime}\mathbf{A}(\mathbf{x}-\mathbf{x}^*) \geqq \mathbf{c}'\mathbf{x}$$

egyenlőtlenség, amiből leolvasható, hogy igaz a

$$z_{\text{opt}} \leqq \mathbf{c}'\mathbf{x}^*+\max_{\mathbf{x}\in\mathbf{D}^n} \lambda^{*\prime}\mathbf{A}(\mathbf{x}-\mathbf{x}^*)$$

felsőbecslés az optimális célfüggvényértékre (ezt jelöli z_{opt}). Az eljárás folyamán azonban mi mindig csak olyan megengedett megoldások iránt érdeklődünk, amelyek jobb célfüggvényértékkel rendelkeznek, mint az eddig talált legmagasabb érték. (Ez utóbbit jelöljük z_0-lal.) Ezért a felső becslés tovább élesíthető:

$$z_{\text{opt}} \leqq \mathbf{c}'\mathbf{x}^*+ \max_{\substack{\mathbf{x}\in\mathbf{D}^n \\ \mathbf{A}\mathbf{x}\leqq\mathbf{b} \\ \mathbf{c}'\mathbf{x}\geqq z_0+\varepsilon}} \lambda^{*\prime}\mathbf{A}(\mathbf{x}-\mathbf{x}^*), \tag{3.3}$$

ahol $\varepsilon\geqq 0$ kicsiny szám.

Legyen $M(\mathbf{x}^*)$ *azon Lagrange-szorzók* halmaza, melyek $\mathbf{x}^*$-ot generálják. Tehát

$$M(\mathbf{x}^*)=\left\{\lambda:\ \begin{cases} \lambda'\mathbf{a}_j \leqq c_j, & \text{ha} \quad x_j^*=1, \\ \lambda'\mathbf{a}_j \geqq c_j, & \text{ha} \quad x_j^*=0, \end{cases} \quad j=1,\ldots,n;\ \lambda\geqq\mathbf{0}\right\},$$

ahol $\mathbf{a}_j$ az $\mathbf{A}$ mátrix oszlopvektorát jelöli. Feltesszük, hogy $M(\mathbf{x}^*)$ nem üres. λ-t úgy kívánjuk megválasztani, hogy a (3.3) becslés (illetőleg annak „folytonos változata") a legpontosabb legyen. Ez azt jelenti, hogy λ a

$$\min_{\lambda\in M(\mathbf{x}^*)} \max_{\substack{\mathbf{x}\geqq\mathbf{0} \\ \mathbf{A}\mathbf{x}\leqq\mathbf{b} \\ \mathbf{c}'\mathbf{x}\geqq z_0+\varepsilon \\ \mathbf{E}\mathbf{x}\leqq\mathbf{e}}} \lambda'\mathbf{A}(\mathbf{x}-\mathbf{x}^*) \tag{3.4}$$

feladat optimális megoldása lesz.

A (3.4) probléma könnyen átírható egy vele ekvivalens lineáris programozási feladattá. Az egyszerűség kedvéért az $\mathbf{x}$ változókra vonatkozó feltételeket a tömörebb

$$\mathbf{T}\mathbf{x}\leqq\mathbf{t}$$

$$\mathbf{x}\geqq\mathbf{0}$$

alakban fogjuk felírni, ahol $\mathbf{T}$, ill. $\mathbf{t}$ a megfelelő $(m+n+1)\times n$-es mátrix, ill. $(m+n+1)$-dimenziós vektor. Tovább alakítva (3.4)-et a

$$\min_{\lambda\in M(\mathbf{x}^*)} \left\{-\lambda'\mathbf{A}\mathbf{x}^*+\max_{\substack{\mathbf{x}\geqq\mathbf{0} \\ \mathbf{T}\mathbf{x}\leqq\mathbf{t}}} \lambda'\mathbf{A}\mathbf{x}\right\}$$

ekvivalens alakot kapjuk. Vegyük észre, hogy a belső maximum probléma minden rögzített λ esetén egy lineáris programozási feladat, vehetjük tehát ennek a duálját. Ekkor a

$$\min_{\lambda \in M(x^*)} \left\{ -\lambda' \mathbf{A}x^* + \min_{\substack{y' \mathbf{T} \geqq \lambda' \mathbf{A} \\ y \geqq 0}} t'y \right\}$$

alakot kapjuk. Mivel $\mathbf{A}x^*$ egy rögzített vektor és a $\lambda \in M(x^*)$ csak lineáris feltételeket tartalmaz, ezért a (3.4)-gyel ekvivalens

$$\min_{\substack{\lambda \in M(x^*) \\ y' \mathbf{T} \geqq \lambda' \mathbf{A} \\ y \geqq 0}} (t'y - \lambda' \mathbf{A}x^*) \tag{3.5}$$

feladat egy lineáris programozási probléma.

4. A Lagrange-szorzók és az s-feltételek kapcsolata

Továbbra is

$$\begin{aligned} &\max \mathbf{c}'\mathbf{x} \\ &\mathbf{A}\mathbf{x} \leqq \mathbf{b} \\ &\mathbf{x} \in \mathbf{D}^n \end{aligned} \tag{4.1}$$

feladattal foglalkozunk, ahol az általánosság megszorítása nélkül feltételezhetjük, hogy $\mathbf{c} \geqq \mathbf{0}$.

Az *s-feltétel* fogalmát GLOVER vezette be [5]-ben. (4.1) feltételeiből egy nemnegatív λ szorzóvektort használva egyetlen feltételt kaphatunk, amely következménye az eredeti rendszernek. (A későbbiekben *ugyanezt* a λ vektort fogjuk használni Lagrange-szorzóként is.) Az *s-feltétel* és a hozzákapcsolódó hátizsákfeladat tehát a következő:

$$\begin{aligned} &\max \mathbf{c}'\mathbf{x} \\ &\lambda'\mathbf{A}\mathbf{x} \leqq \lambda'\mathbf{b} \\ &\mathbf{x} \in \mathbf{D}^n. \end{aligned} \tag{4.2}$$

Az egyszerűség kedvéért a továbbiakban az *s-feltételt* az

$$\mathbf{a}'\mathbf{x} \leqq d$$

alakban fogjuk használni. Itt eltekintünk az egyébként érdektelen $a_j = c_j = 0$ esettől.

4.1. DEFINÍCIÓ: A változók *s-feltétel* szerinti rendezése. x_i megelőzi x_j-t (jelölésben $x_i < x_j$), ha az alábbi két eset valamelyike teljesül:

a) $a_i \leqq 0$ és $a_j > 0$,

b) $a_i, a_j > 0$ és $\dfrac{c_i}{a_i} > \dfrac{c_j}{a_j}$.

Ez a rendezés az *s-feltételtől*, tehát végső soron a λ szorzóvektortól függ. (4.2) egy hátizsákfeladat, és a most definiált konstrukció az ott szokásos rendezésnek felel meg.

Ha $x_i \prec x_j$ teljesül, akkor

$$w_1 = \frac{c_i}{a_i} > \frac{c_j}{a_j} = w_2.$$

Legyen

$$w^* = \frac{w_1 + w_2}{2}.$$

Most $\bar{x}_i(w^*)=1$ és $\bar{x}_j(w^*)=0$. Bebizonyítottuk tehát, hogy ha az $x_i \prec x_j$ reláció igaz, akkor $x_i \lhd x_j$ is fennáll.

Megfordítva, tegyük fel, hogy $x_i \lhd x_j$. Ez azt jelenti, hogy létezik olyan $\bar{w}$, amelyre

$$c_i - \bar{w}a_i \geqq 0 > c_j - \bar{w}a_j.$$

A fentiek alapján innen azonnal következik, hogy $a_j > 0$. A következő két eset lehetséges:

a) $a_i \leqq 0$ és ezért $x_i \prec x_j$ vagy

b) $a_i > 0$ és ekkor

$$\frac{c_i}{a_i} \geqq \bar{w} > \frac{c_j}{a_j}$$

és így $x_i \prec x_j$.

Ezzel a lemma bizonyítását befejeztük.

A most igazolt állításból több következtetés vonható le:

1. Egy olyan, a *Lagrange-szorzók* használatára alapozott algoritmus, mint amilyennek az általános lépését az 1. szakaszban fogalmaztuk meg, az *s-feltétel*lel definiált (4.2) feladatnak közelítő megoldásait szolgáltatja.

2. *Lagrange-szorzók*ra épített eljárásban figyelembe vehetjük az *s-feltétel* jobboldalát, hogy (4.2), illetve (4.1) megengedett megoldásaihoz jussunk. Ha

$$\mathbf{a}'\bar{\mathbf{x}} > d,$$

akkor az 1 szinten levő változók közül a rendezésben az utolsóknak az értékeit 0-ra változtatjuk mindaddig, amíg megengedett megoldáshoz nem jutunk. Ez az eljárás a következőt jelenti. $\bar{\mathbf{x}}$ meghatározásakor a (4.3) feladatot oldottuk meg $w=1$ értékkel. Most w-t növeljük, ezáltal egyes ismeretlenek értéke 0-ra csökken. Ez viszont azt jelenti, hogy az így kapott pontokat továbbra is *Lagrange-szorzók* generálják, tehát megtartják az ilyen vektorok összes tulajdonságát. Miért fontos mindez? R. BROOKS és A. GEOFFRION [2]-ben megadtak egy szabályt a szorzók választására. Ennek az egyébként jó módszernek nagy a „tehetetlensége", ami azt jelenti, hogy addig nem tud szorzókat adni, amíg megengedett megoldást nem ismerünk. Tehát mielőtt a szorzók választásának erre a módjára rátérhetünk, előbb egy másik eljárást kell követni. Ez lehet például a véletlenszerű választás. A számítógépes tapasztalatok azt mutatják, hogy ilyen esetben sokkal könnyebben jutunk megengedett megoldáshoz, ha használjuk a *Lagrange-szorzócsaládra* vonatkozó ismereteinket. Ennek oka az, hogy míg a szorzók egymás közti arányai jók, nagyságuk a rögzített célfüggvényhez viszonyítva nincs helyesen beállítva.

3. A (4.1) feladatnak egy *Lagrange-szorzók* által generált megengedett megoldása optimális is, ha optimális megoldása (4.2)-nek. Sajnos azonban ma még nem rendelkezünk elég jó tesztekkel arra vonatkozóan, hogy egy heurisztikus módon előállított megoldás optimális-e egy hátizsák feladatban, vagy sem.

4.2. DEFINÍCIÓ: A λ vektor által meghatározott *Lagrange-szorzócsalád* a

$$\{w\lambda: w > 0\}$$

halmaz.

4.3. DEFINÍCIÓ: A változók rendezése egy *Lagrange-szorzócsalád* szerint. Tekintsük a család tagjaival felírt *Lagrange-feladat*okat:

$$\max_{\mathbf{x}\in D^n} [\mathbf{c}'\mathbf{x} - w\lambda'\mathbf{A}\mathbf{x}] = \max_{\mathbf{x}\in D^n} [\mathbf{c}'\mathbf{x} - w\mathbf{a}'\mathbf{x}]. \tag{4.3}$$

Jelölje $\bar{\mathbf{x}}(w)$ (4.3) azon optimális megoldását, amelyben egyetlen változó értékét sem lehet 0-ról 1-re változtatni az optimalitás elvesztése nélkül. Ekkor x_i megelőzi x_j-t $(i \neq j)$ (jelölésben $x_i \lhd x_j$), ha létezik olyan w^*, hogy

$$\bar{x}_i(w^*) = 1 \quad \text{és} \quad \bar{x}_j(w^*) = 0.$$

Be kell bizonyítanunk, hogy az így definiált fogalom valóban egy rendezés. A következő lemma bizonyításából látható, hogy ha $x_i \lhd x_j$, akkor nincs olyan $\tilde{w}$, hogy $\bar{x}_i(\tilde{w}) = 0$ és $\bar{x}_j(\tilde{w}) = 1$.

Ezt a tényt felhasználva megmutatjuk, hogy a „$\lhd$" reláció tranzitív. Legyen ugyanis i, j, k három különböző index és $x_i \lhd x_j$, $x_j \lhd x_k$, ekkor létezik egy $\hat{w}$, hogy

$$\bar{x}_j(\hat{w}) = 1 \quad \text{és} \quad \bar{x}_k(\hat{w}) = 0. \tag{4.4}$$

Most

$$\bar{x}_i(\hat{w}) = 1, \tag{4.5}$$

különben $x_j \lhd x_i$ is fennállna. (4.4) és (4.5) együtt pedig azt jelenti, hogy $x_i \lhd x_k$.

Könnyen látható, hogy egyik reláció sem teljes rendezés.

4.1. LEMMA: A (4.1) feladat változóinak a λ vektor által generált s-feltétel szerinti, ill. a λ vektor által meghatározott *Lagrange-szorzócsalád* szerinti rendezései megegyeznek egymással.

Bizonyítás. Oldjuk meg a (4.3) feladatot. Ha

$$c_j - wa_j \geqq 0,$$

akkor $\bar{x}_j = 1$ és

$$c_j - wa_j < 0$$

esetén $\bar{x}_j = 0$. Mivel a $c_i = a_i = 0$ esetet kizártuk és $w > 0$ ezért $a_i \leqq 0$ esetén

$$c_i - wa_i > 0.$$

Ez azt jelenti, hogy minden olyan x_i, amelyre $a_i \leqq 0$ megelőzi a „$\lhd$" reláció szerint azokat az x_j-ket, amelyekre $a_j > 0$, hiszen az utóbbi esetben van olyan $\bar{w}$, hogy

$$c_j - \bar{w}a_j < 0$$

teljesül.

Tekintsük most az $a_i, a_j > 0$ esetet. $\bar{x}_i(w) = 1$ azt jelenti, hogy

$$c_i - wa_i \geqq 0.$$

Innen

$$\frac{c_i}{a_i} \geqq w.$$

IRODALOM

[1] BALAS, E., "An additive algorithm for solving linear programs with zero-one variables", *Operations Research* **13** (1965) 517—546.
[2] BROOKS, R. and GEOFFRION, A., "Finding Everett's Lagrange multipliers by linear programming", *Operations Research* **14** (1966) 1149—1153.
[3] EVERETT, H. III. "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources", *Operations Research* **11** (1963) 399—417.
[4] FISHER, M. L. and SHAPIRO, J. F., "Constructive duality in integer programming", *SIAM J. Appl. Math.* 1974. 31—52.
[5] GLOVER, F., "A multiphase-dual algorithm for the zero-one integer programming problem", *Operations Research* **13** (1965) 879—919.
[6] GLOVER, F., "Surrogate constraint duality in mathematical programming", *Operations Research* **23** (1975) 434—451.
[7] KAPLAN, S., "Solution of the Loire-Savage and similar integer programming problems by generalized Lagrange multiplier method", *Operations Research* **14** (1966) 1130—1136.
[8] KOVÁCS, L. B., „Leszámlálási struktúrák és alkalmazásuk diszkrét programozási feladatok megoldására", *Matematikai Lapok* **XIX** (1968) 33—48.
[9] KOVÁCS, L. B., *A diszkrét programozás kombinatorikus módszerei* (Bolyai János Matematikai Társulat, Budapest, 1969).
[10] LASDON, L. S., *Optimization Theory for Large Systems* (The MacMillan Company, London, 1971).
[11] NEMHAUSER, G. L. and ULLMANN, Z., "A note on the generalized Lagrange multipliers solution to an integer programming problem", *Operations Research* **16** (1968) 450—453.
[12] SHAPIRO, J. F., "Dynamic programming algorithms for the integer programming problem I.", *Operations Research* **16** (1968) 103—121.
[13] SCHWEITZER, P. J., "Optimization with an approximate Lagrangian", *Mathematical Programming* **5** (1974) 191—198.
[14] VIZVÁRI, B., „Egy diszkrét programozási feladat megoldása *Lagrange-szorzókkal*", Szakdolgozat, Eötvös Loránd Tudományegyetem, Budapest, 1973.
[15] WASHBURN, A., "A note on integer maximization of unimodal functions", *Operations Research* **23** (1975) 358—360.
[16] WHITE, D. J., "Dynamic programming and probabilistic constraints", *Operations Research* **21** (1973) 654—664.

(Beérkezett: 1977. július 29.)

VIZVÁRI BÉLA
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1111 BUDAPEST XI. KENDE U. 13—17.

THE USE OF LAGRANGEAN MULTIPLIERS IN INTEGER PROGRAMMING ALGORITHMS

B. VIZVÁRI

The aim of the present paper is to show how the generalized *Lagrange multipliers method*, as it was suggested by H. EVERETT, can be used in an enumeration algorithm. First of all we give a new optimality test. Though the test is a general one, its use in integer programming is detailed. We want to accelerate the second phase of the enumeration algorithms which is the proof of the optimality of the best feasible solution. The existence of gaps is the greatest difficulty of the GLM method. We show how it can be avoided in integer programming. In the last section the relation between the Lagrange multipliers and the surrogate constraints is examined.

Alkalmazott Matematikai Lapok 2 (1976) 427—437

A SZEKVENCIÁLIS, FELTÉTEL NÉLKÜLI MINIMALIZÁLÁSI MÓDSZER (SUMT) ALKALMAZÁSA NEM KONVEX PROGRAMOZÁSI FELADATOK ESETÉN

RAPCSÁK TAMÁS
Budapest

Ebben a dolgozatban azt vizsgáljuk, hogy milyen feltételek mellett tudjuk eldönteni, hogy egyenlőtlenség formában megadott feltételek esetén a büntetőfüggvények lokális optimumainak egy sorozata mikor konvergál a feladat globális optimumához. Először a kvázikonvex és a pszeudokonvex függvények tulajdonságaival, majd a globális optimalitás első és másodrendű, szükséges és elegendő feltételeivel foglalkozunk. E vizsgálatok segítségével, — Kuhn—Tucker pontot keresve — tudjuk végül is eldönteni a SUMT módszer alkalmazása esetén, hogy globális vagy lokális optimumot találtunk-e. Ezután a konvex függvényeknél általánosabb függvények esetén egy, a globális optimumot megadó SUMT algoritmust ismertetünk.

1. Bevezetés

Jólismert tény az, hogy ha egy olyan nemlineáris programozási problémát tekintünk, amely nem konvex programozási feladat, akkor a SUMT-módszer alkalmazásakor nem tudjuk azt eldönteni, hogy lokális vagy globális optimumot kaptunk-e. Ebben a dolgozatban azt vizsgáljuk, hogy milyen feltételek mellett tudjuk eldönteni, hogy a büntetőfüggvények lokális optimumainak egy sorozata mikor konvergál a feladat globális optimumához. A dolgozat végén konvex függvényeknél általánosabb függvények esetére adunk globális optimumot meghatározó algoritmust. Itt felhasználjuk GERENCSÉR ([6]) tételét.

A dolgozat első részében a kvázikonvex, a pszeudokonvex és a szigorúan pszeudokonvex függvények közötti kapcsolatot vizsgáljuk.

A dolgozat második részében a nemlineáris programozásban gyakran alkalmazott, az optimalitás úgynevezett „másodrendű szükséges és elegendő" feltételeinek teljesülésére adunk új feltételeket.

A következő részben vizsgáljuk azt a kérdést, hogy hogyan tudjuk a SUMT-módszer alkalmazása esetén eldönteni, hogy a büntető függvények lokális optimumainak egy sorozata mikor konvergál a feladat globális optimumához.

Végül egy olyan SUMT-algoritmust ismertetünk, amely a konvex függvényeknél általánosabb függvények esetén a *Kuhn—Tucker-féle feltételek* teljesülését vizsgálva ad globális optimumot.

2. Kvázikonvex, pszeudokonvex és szigorúan pszeudokonvex függvények

Ebben a részben a fenti függvények közötti kapcsolatot vizsgáljuk. Itt ismert állítások is szerepelnek, amelyeket egyszerűbb bizonyításokkal közlünk.

2.1. DEFINÍCIÓ. Egy $f(\mathbf{x})$ numerikus függvényt, amely definiálva van egy $\Gamma \subset R^n$ halmazon, egy $\bar{\mathbf{x}} \in \Gamma$ pontban kvázikonvexnek nevezünk (a Γ halmazra vonatkoztatva), ha bármely $\mathbf{x} \in \Gamma$ esetén, amelyre $f(\mathbf{x}) \leqq f(\bar{\mathbf{x}})$, az $f(\mathbf{x})$ értéke az $[\bar{\mathbf{x}}, \mathbf{x}]$ zárt szakasz és Γ metszetén kisebb, vagy egyenlő, mint $f(\bar{\mathbf{x}})$.

Ez nem más, mint hogy

$$\left.\begin{aligned} \mathbf{x} &\in \Gamma \\ f(\mathbf{x}) &\leqq f(\bar{\mathbf{x}}) \\ 0 \leqq \lambda &\leqq 1 \\ (1-\lambda)\bar{\mathbf{x}} + \lambda\mathbf{x} &\in \Gamma \end{aligned}\right\} \Rightarrow f[(1-\lambda)\bar{\mathbf{x}} + \lambda\mathbf{x}] \leqq f(\bar{\mathbf{x}}).$$

Az $f(\mathbf{x})$ kvázikonvex a Γ halmazon, ha kvázikonvex bármely $\mathbf{x} \in \Gamma$ esetén.

2.2. DEFINÍCIÓ. Egy $f(\mathbf{x})$ numerikus függvény, amely definiálva van egy $\Gamma \subset R^n$ halmazon, egy $\bar{\mathbf{x}} \in \Gamma$ pontban szigorúan kvázikonvex (a Γ halmazra vonatkoztatva), ha

$$\left.\begin{aligned} \mathbf{x} &\in \Gamma \\ f(\mathbf{x}) &< f(\bar{\mathbf{x}}) \\ 0 < \lambda &< 1 \\ (1-\lambda)\bar{\mathbf{x}} + \lambda\mathbf{x} &\in \Gamma \end{aligned}\right\} \Rightarrow f[(1-\lambda)\bar{\mathbf{x}} + \lambda\mathbf{x}] < f(\bar{\mathbf{x}}).$$

Az $f(\mathbf{x})$ szigorúan kvázikonvex a Γ halmazon, ha szigorúan kvázikonvex bármely $\mathbf{x} \in \Gamma$ esetén. Ha az $f(\mathbf{x})$ szigorúan kvázikonvex és folytonos a Γ konvex halmazon, akkor ott kvázikonvex [11].

2.3. DEFINÍCIÓ. Egy $f(\mathbf{x})$ numerikus függvény, amely definiálva van egy $\Gamma \subset R^n$ halmazon, egy $\bar{\mathbf{x}} \in \Gamma$ pontban pszeudokonvex (a Γ halmazra vonatkoztatva), ha $f(\mathbf{x})$ differenciálható az $\bar{\mathbf{x}}$ pontban és ha

$$\left.\begin{aligned} \mathbf{x} &\in \Gamma \\ \nabla f(\bar{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}}) &\geqq 0 \end{aligned}\right\} \Rightarrow f(\mathbf{x}) \geqq f(\bar{\mathbf{x}}).$$

(A $\nabla f(\mathbf{x})$ sorvektort fog jelenteni a továbbiakban is.) Az $f(\mathbf{x})$ pszeudokonvex a Γ halmazon, ha pszeudokonvex bármely $\mathbf{x} \in \Gamma$ esetén.

2.4. DEFINÍCIÓ. Egy $f(\mathbf{x})$ numerikus függvény, amely definiálva van egy $\Gamma \subset R^n$ halmazon, egy $\bar{\mathbf{x}} \in \Gamma$ pontban szigorúan pszeudokonvex (a Γ halmazra vonatkoztatva), ha $f(\mathbf{x})$ differenciálható az $\bar{\mathbf{x}}$ pontban és ha

$$\left.\begin{aligned} \mathbf{x} &\in \Gamma \\ \nabla f(\bar{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}}) &\geqq 0 \end{aligned}\right\} \Rightarrow f(\mathbf{x}) > f(\bar{\mathbf{x}}).$$

Az $f(\mathbf{x})$ szigorúan pszeudokonvex a Γ halmazon, ha szigorúan pszeudokonvex bármely $\mathbf{x} \in \Gamma$ esetén.

A következőkben legyen $f(\mathbf{x})$ egy numerikus függvény, amely definiálva van egy $\Gamma \subset R^n$ nyílt, konvex halmazon, és legyen $\mathbf{x} \in \Gamma$.

Tekintsük az $S = \{\mathbf{x} | f(\mathbf{x}) = f(\mathbf{x}_0)\}$ halmazt, illetve az $S_1 = \{\mathbf{x} | f(\mathbf{x}) \leqq f(\mathbf{x}_0)\}$ halmazt.

2.5. LEMMA [11]. Ha az $f(\mathbf{x})$ differenciálható az $\mathbf{x}_0$ pontban, és $f(\mathbf{x})$ kvázikonvex az $\mathbf{x}_0$ pontban, akkor $f(\mathbf{x}) \leqq f(\mathbf{x}_0) \Rightarrow \nabla f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) \leqq 0$.

Ez az állítás geometriailag azt jelenti, hogy ha $\nabla f(\mathbf{x}_0) \neq \mathbf{0}$ és az S halmaz egy felületet határoz meg, akkor az S felület $\mathbf{x}_0$ pontbeli érintősíkja egyben az S_1 halmaz támaszsíkja is.

2.6. LEMMA. Ha az $f(\mathbf{x})$ folytonos a Γ halmazon, az $\mathbf{x}_0$ pontban differenciálható és kvázikonvex, $\nabla f(\mathbf{x}_0) \neq \mathbf{0}$, akkor

$$\nabla f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) \geqq 0 \Rightarrow f(\mathbf{x}) \geqq f(\mathbf{x}_0).$$

Bizonyítás. Tegyük fel az állítással ellentétben, hogy $\exists \hat{\mathbf{x}}$, amelyre

$$\left.\begin{aligned} \nabla f(\mathbf{x}_0)(\hat{\mathbf{x}} - \mathbf{x}_0) \geqq 0 \\ \hat{\mathbf{x}} \in \Gamma \end{aligned}\right\} \Rightarrow f(\hat{\mathbf{x}}) < f(\mathbf{x}_0).$$

Ez az $f(\mathbf{x})$ folytonossága miatt azt jelenti, hogy az $\hat{\mathbf{x}}$ pont egy $Q(\hat{\mathbf{x}}, \delta)$ $(\delta > 0)$ környezete esetén is

$$f(\mathbf{x}) < f(\mathbf{x}_0), \quad \mathbf{x} \in Q(\hat{\mathbf{x}}, \delta).$$

Másrészt a 2.5. lemma miatt

$$\nabla f(\mathbf{x}_0)(\hat{\mathbf{x}} - \mathbf{x}_0) = 0$$

azaz $\hat{\mathbf{x}}$ az S_1 halmaz támasztósíkjában van.

Ez ellentmondás, mert az előbbiek szerint a támaszsík mindkét oldalán van S_1 halmazhoz tartozó pont.

Az előbbi lemmához hasonló állításokat találunk [3], [4]-ben.

2.7. KÖVETKEZMÉNY. Ha az $f(\mathbf{x})$ folytonos a Γ halmazon, az $\mathbf{x}_0$ pontban differenciálható és kvázikonvex, $\nabla f(\mathbf{x}_0) \neq \mathbf{0}$, akkor $f(\mathbf{x})$ az $\mathbf{x}_0$ pontban pszeudokonvex.

Mivel egy konvex halmazon pszeudokonvex függvény ott szigorúan kvázikonvex is, így az előbbi lemma feltételei egyben a szigorú kvázikonvexitásnak is elegendő feltételei.

2.8. LEMMA. Ha az $f(\mathbf{x})$ kétszer folytonosan differenciálható a Γ halmazon és egy $\mathbf{x}_0$ pontban kvázikonvex és $\nabla f(\mathbf{x}_0) \neq \mathbf{0}$, akkor

$$\mathbf{v}^T \nabla^2 f(\mathbf{x}_0) \mathbf{v} \geqq 0, \quad \text{ha} \quad \nabla f(\mathbf{x}_0) \mathbf{v} = 0.$$

(Az $f(\mathbf{x})$ függvény *Hesse-mátrix*át az $\mathbf{x}_0$ pontban a $\nabla^2 f(\mathbf{x}_0)$ szimbólum jelöli.)

Bizonyítás. Legyen $\mathbf{x}_1$ egy olyan pont, amelyre $\nabla f(\mathbf{x}_0)(\mathbf{x}_1 - \mathbf{x}_0) = 0$ és legyen $\tilde{\mathbf{x}}_1 = \mathbf{x}_0 + \lambda(\mathbf{x}_1 - \mathbf{x}_0)$, $0 \leqq \lambda \leqq 1$. Így

$$f(\tilde{\mathbf{x}}_1) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0) \lambda(\mathbf{x}_1 - \mathbf{x}_0) + \frac{1}{2} \lambda(\mathbf{x}_1 - \mathbf{x}_0)^T \nabla^2 f(\mathbf{x}_0 + \theta\lambda(\mathbf{x}_1 - \mathbf{x}_0)) \lambda(\mathbf{x}_1 - \mathbf{x}_0),$$

$$0 < \theta < 1. \tag{2.1}$$

Mivel az $f(\mathbf{x})$ az $\mathbf{x}_0$ pontban pszeudokonvex, ezért

$$\nabla f(\mathbf{x}_0)(\mathbf{x}_1-\mathbf{x}_0)=0 \Rightarrow f(\mathbf{x}_1)\geqq f(\mathbf{x}_0) \Rightarrow$$

$$\Rightarrow (\mathbf{x}_1-\mathbf{x}_0)^T\nabla^2 f(\mathbf{x}_0+\theta\lambda(\mathbf{x}_1-\mathbf{x}_0))(\mathbf{x}_1-\mathbf{x}_0)\geqq 0$$

azaz

$$(\mathbf{x}_1-\mathbf{x}_0)^T\nabla^2 f(\mathbf{x}_0)(\mathbf{x}_1-\mathbf{x}_0)=\lim_{\lambda\to 0}(\mathbf{x}_1-\mathbf{x}_0)^T\nabla^2 f(\mathbf{x}_0+\theta\lambda(\mathbf{x}_1-\mathbf{x}_0))(\mathbf{x}_1-\mathbf{x}_0)\geqq 0,$$

ami éppen az állítás.

A 2.8 lemma állítása más formában megtalálható [2], [6]-ban.

Legyen $\mathbf{v}\neq\mathbf{0}$ egy érintővektor, azaz $\nabla f(\mathbf{x}_0)\mathbf{v}=0$. Vizsgáljuk azt a görbét, amelyet a $\mathbf{v}, \nabla f(\mathbf{x}_0)$ vektorok által kifeszített sík metsz ki az $S=\{\mathbf{x}|f(\mathbf{x})=f(\mathbf{x}_0)\}$ felületből. Ha ezt a görbét a $\mathbf{v}, \nabla f(\mathbf{x}_0)$ vektorok által felfeszített koordinátarendszerben tekintjük, $u(t)$ jelöli és $u(0)=\mathbf{x}_0$, akkor a $\nabla f(\mathbf{x}_0)\nabla f(\mathbf{x}_0)^T\neq 0$ reláció miatt ezt a görbét az alábbi egyenlet definiálja.

$$f(\mathbf{x}_0+t\mathbf{v}+u(t)\nabla f(\mathbf{x}_0))=f(\mathbf{x}_0). \tag{2.2}$$

Legyen $\frac{du}{dt}=\dot{u}(t)$.

2.9. LEMMA [6]. Ha $\|\nabla f(\mathbf{x}_0)\|=1$, akkor

$$\mathbf{v}^T\nabla^2 f(\mathbf{x}_0)\mathbf{v}=-\ddot{u}(0), \quad \text{ha} \quad \nabla f(\mathbf{x}_0)\mathbf{v}=0, \quad \mathbf{v}\neq\mathbf{0}.$$

Bizonyítás. Differenciáljuk kétszer a (2.2) egyenlőség mindkét oldalát t szerint, majd a kapott eredményt tekintsük a $t=0$ pontban. Így kapjuk, hogy

$$\nabla f[\mathbf{x}_0+t\mathbf{v}+u(t)\nabla f(\mathbf{x}_0)](\mathbf{v}+\dot{u}(t)\nabla f(\mathbf{x}_0)^T)=0, \tag{2.3}$$

$$\begin{aligned}(\mathbf{v}+\dot{u}(t)\nabla f(\mathbf{x}_0)^T)^T\nabla^2 f[\mathbf{x}_0+t\mathbf{v}+u(t)\nabla f(\mathbf{x}_0)](\mathbf{v}+\dot{u}(t)\nabla f(\mathbf{x}_0)^T)+\\ +\ddot{u}(t)\nabla f(\mathbf{x}_0)\nabla f[\mathbf{x}_0+t\mathbf{v}+u(t)\nabla f(\mathbf{x}_0)]^T=0.\end{aligned} \tag{2.4}$$

Mivel $\dot{u}(0)=0$, így azt kapjuk a második egyenlőségből, hogy

$$\mathbf{v}^T\nabla^2 f(\mathbf{x}_0)\mathbf{v}=-\ddot{u}(0),$$

ami éppen az állítás.

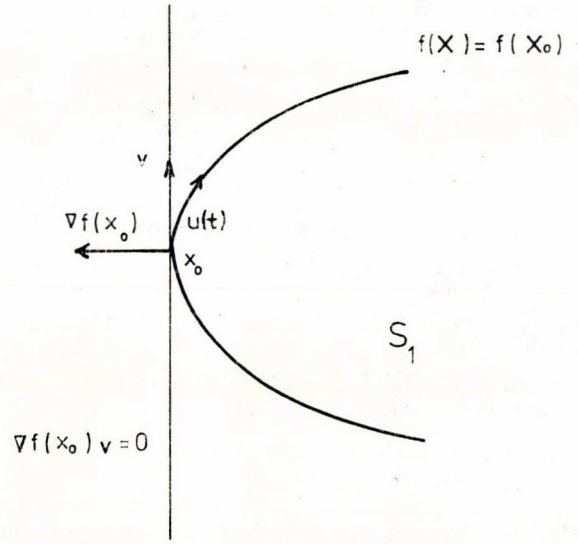
Ha az S_1 halmaz konvex akkor az $u(t)$ görbe egy darabon konkáv, így ebből a szemléletes tényből a 2.9. lemma segítségével azonnal adódik a 2.8. lemma állítása. Ezt látjuk az 1. ábrán.

Most a szigorú pszeudokonvexitásra adunk egy a korábbitól eltérő definíciót. Ez a definíció [6]-ban már szerepelt, mint a szigorú kvázikonvexitás definíciója.

A (2.4) definíció értelmében az $\mathbf{x}_0$ pontbeli szigorú pszeudokonvexitás geometriailag azt jelenti, hogy minden lehetséges $\mathbf{v}\neq\mathbf{0}$ érintőirány esetén a megfelelő $u(t)$ görbének az $\mathbf{x}_0$ pontban szigorú, maximuma van. Erre nézve egy elegendő feltétel az, hogy az $\ddot{u}(0)<0$ egyenlőtlenség teljesüljön minden $u(t)$ görbe esetén.

2.10. DEFINÍCIÓ. Egy $f(\mathbf{x})$ numerikus függvény, amely definiálva van egy $\Gamma\subset R^n$ nyílt halmazon, egy $\bar{\mathbf{x}}\in\Gamma$ pontban szigorúan pszeudokonvex a Γ halmazra nézve, ha $f(\mathbf{x})$ kétszer folytonosan differenciálható a Γ-n, az $\bar{\mathbf{x}}$ pontban kvázikonvex és

$$\mathbf{v}^T\nabla^2 f(\bar{\mathbf{x}})\mathbf{v}>0, \quad \text{ha} \quad \nabla f(\bar{\mathbf{x}})\mathbf{v}=0, \quad \nabla f(\bar{\mathbf{x}})\neq\mathbf{0}, \quad \mathbf{v}\neq\mathbf{0}.$$



1. ábra

Ez a definíció kvadratikus függvények esetén ugyanazt jelenti, mint a 2.4. definíció.Ezt a (2.1) formula segítségével azonnal beláthatjuk. Általános esetben is a két definíció kétszer folytonosan differenciálható függvények esetén néhány kivételtől eltekintve ugyanazt a függvényhalmazt fedi le. Azonban mint látni fogjuk, ilyen módon a szigorúan pszeudokonvex függvények analitikusan igen jól kezelhetők.

3. Az optimalitás szükséges és elegendő feltételei

A következő feladattal foglalkozunk:

$$\min f(\mathbf{x}), \qquad g_i(\mathbf{x}) \geqq 0, \quad i = 1, \ldots, m, \tag{3.1}$$

ahol az $f(\mathbf{x})$, $g_i(\mathbf{x})$, $i=1, \ldots, m$ függvények egy $\Gamma \subset R^n$ nyílt halmazon értelmezett folytonos függvények.

Legyen az $\mathbf{x}^*$ pont a (3.1) feladat megoldása és legyenek az $f(\mathbf{x})$, $g_i(\mathbf{x})$, $i=1, \ldots, m$ függvények az $\mathbf{x}^*$ pontban differenciálhatók. Jelölje R és R_0 az alábbi halmazokat.

$$R = \{\mathbf{x} | g_i(\mathbf{x}) \geqq 0,\ i = 1, \ldots, m,\ \mathbf{x} \in \Gamma\}, \tag{3.2}$$

$$R_0 = \{\mathbf{x} | g_i(\mathbf{x}_0) > 0,\ i = 1, \ldots, m,\ \mathbf{x} \in \Gamma\}, \quad P = R/R_0. \tag{3.3}$$

Tételezzük fel a továbbiakban, hogy az R halmaz konvex.

Akkor mondjuk, hogy az R tartomány egy $\mathbf{x}$ pontjában $B(\mathbf{x})$ az aktív indexek halmaza, ha

$$g_i(\mathbf{x}) = 0, \quad i \in B(\mathbf{x}), \quad g_i(\mathbf{x}) > 0, \quad i \notin B(\mathbf{x}).$$

A most következő részben először a globális optimalitásra vonatkozó szükséges és elegendő feltételeket mondjuk ki, (a *Kuhn—Tucker-féle feltételek* segítségével), amely már figyelembe veszi a választott algoritmus sajátosságait is.

Ebben a tételben az alábbi regularitási feltételt használjuk.

Ha $\mathbf{x}^* \in R$ a feladat megoldása, $B(\mathbf{x}^*)$ az aktív indexek halmaza, akkor azt feltételezzük, hogy tudunk találni olyan $\mathbf{y}$ R-beli vektort, hogy a

$$\nabla g_i(\mathbf{x}^*)\,(\mathbf{y}-\mathbf{x}) > 0, \quad i \in B(\mathbf{x}^*) \tag{3.4}$$

egyenlőtlenségek teljesüljenek.

Ez a regularitási feltétel megtalálható az [5], [14], [15] dolgozatokban. A (3.4) egyenlőtlenségek teljesülése konkáv függvények által meghatározott (3.2) tartomány esetén ekvivalens az $R_0 \neq \emptyset$ feltétellel. Most a (3.1) feladat esetén próbálunk a (3.4) feltétel teljesülésére könnyebben ellenőrizhető feltételeket adni.

3.1. LEMMA. Ha az R halmaz konvex, $R_0 \neq \emptyset$ és $\mathbf{x}^* \in P$ esetén $\nabla g_i(\mathbf{x}^*) \neq \mathbf{0}$, $i \in B(\mathbf{x}^*)$, akkor az $\mathbf{x}^*$ pontban a (3.4) feltétel teljesül.

Bizonyítás. Először belátjuk azt, hogy ha $\mathbf{x}' \in R$ egy tetszőleges pont, akkor

$$\nabla g_i(\mathbf{x}^*)(\mathbf{x}'-\mathbf{x}^*) \geqq 0, \quad i \in B(\mathbf{x}^*). \tag{3.5}$$

Mivel az R konvex halmaz, ezért

$$\tilde{\mathbf{x}} = \mathbf{x}^* + \lambda(\mathbf{x}'-\mathbf{x}^*) \in R, \quad \text{ha} \quad \mathbf{x}' \in R, \quad 0 \leqq \lambda \leqq 1.$$

Így

$$g_i(\tilde{\mathbf{x}}) - g_i(\mathbf{x}^*) \geqq 0, \quad i \in B(\mathbf{x}^*)$$

és

$$g_i(\tilde{\mathbf{x}}) - g_i(\mathbf{x}^*) = \lambda \nabla g_i[\mathbf{x}^* + \theta_i \lambda(\mathbf{x}'-\mathbf{x}^*)](\mathbf{x}'-\mathbf{x}^*), \quad 0 < \theta_i < 1, \quad i \in B(\mathbf{x}^*). \tag{3.6}$$

Ha most $\lambda \to 0$, akkor éppen a kívánt (3.5) egyenlőtlenségeket kapjuk. Ezután a lemma állítását indirekt úton bizonyítjuk. Tegyük fel, hogy a (3.4) feltétel nem teljesül az $\mathbf{x}^*$ pontban. Ez azt jelenti, hogy legalább egy $\bar{\imath} \in B(\mathbf{x}^*)$ index esetén a

$$\nabla g_{\bar{\imath}}(\mathbf{x}^*)(\mathbf{z}-\mathbf{x}^*) \leqq 0, \quad \bar{\imath} \in B(\mathbf{x}^*), \quad \mathbf{z} \in R \tag{3.7}$$

egyenlőtlenség teljesül. Az $\bar{\imath}$ index a $\mathbf{z}$-től függően változhat.

Ha $\mathbf{y} \in R_0$, akkor a (3.5) és (3.7) relációk miatt

$$\nabla g_{\bar{\imath}}(\mathbf{x}^*)(\mathbf{y}-\mathbf{x}^*) = 0. \tag{3.8}$$

A (3.5) egyenlőtlenségek miatt a (3.8) egyenlőség azt jelenti, hogy az $\mathbf{y}-\mathbf{x}^*$ vektor az R halmaz $\nabla g_{\bar{\imath}}(\mathbf{x}^*)$ normálisú támaszsíkjában van. Mivel $\mathbf{y} \in R_0$, ezért ez azt jelenti, hogy a támaszsík mindkét oldalán van R halmazbeli pont, ami ellentmondás.

3.2. TÉTEL. Ha a (3.1) problémát tekintjük, $R_0 \neq \emptyset$ és egy $\mathbf{x}^* \in R$ pontban az $f(\mathbf{x})$ kvázikonvex, (a Γ halmazra nézve) $\nabla f(\mathbf{x}^*) \neq \mathbf{0}$ $\nabla g_i(\mathbf{x}^*) \neq \mathbf{0}$, $i \in B(\mathbf{x}^*)$, akkor annak szükséges és elegendő feltétele, hogy az $\mathbf{x}^*$ pont a (3.1) probléma globális

minimuma legyen az, hogy a *Kuhn—Tucker-féle feltételek* teljesüljenek, azaz létezzen egy $\mathbf{u}^*$ vektor úgy, hogy

$$\nabla f(\mathbf{x}^*) - \sum_{i=1}^{m} u_i^* \nabla g_i(\mathbf{x}^*) = 0, \tag{3.9}$$

$$u_i^* g_i(\mathbf{x}^*) = 0, \quad i = 1, \ldots, m, \tag{3.10}$$

$$g_i(\mathbf{x}^*) \geqq 0, \quad i = 1, \ldots, m, \tag{3.11}$$

$$u_i^* \geqq 0, \quad i = 1, \ldots, m. \tag{3.12}$$

A tétel bizonyítása mindkét irányban a (3.5) egyenlőtlenség felhasználásával egyszerűen elvégezhető, ezért nem részletezzük.

Megjegyezzük, hogy a 3.2. tételben szereplő $R_0 \neq \emptyset$ feltétel a SUMT belső pont algoritmusaihoz is szükséges, hiszen egy ilyen tulajdonságú pont lesz az induló pont, másrészt a $\nabla f(\mathbf{x}^*) \neq \mathbf{0}$, $\nabla g_i(\mathbf{x}^*) \neq \mathbf{0}$, $i \in B(\mathbf{x}^*)$ feltételek egyszerűen ellenőrizhetők.

Könnyen látható az is, hogy a SUMT módszer konvergencia bizonyításainál felhasznált $\bar{R}_0 = R$ feltétel (az $\bar{R}_0$ szimbólum az R_0 halmaz lezártját jelenti) helyettesíthető a (3.4) és az $R_0 \neq \emptyset$ feltételekkel, ugyanis a konvergencia bizonyításoknál csak azt használjuk fel, hogy az optimum pont egy tetszőlegesen kicsiny környezetében van R_0-beli pont.

A következőkben a (3.1) probléma $\mathbf{x}^*$ megoldására vonatkoztatva az optimalitás másodrendű feltételeivel foglalkozunk, ezért feltételezzük, hogy a (3.1) problémában szereplő függvények kétszer folytonosan differenciálhatók a Γ halmazon.

Vezessük be az alábbi jelölést:

$$L(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}) - \sum_{i=1}^{m} u_i g_i(\mathbf{x}), \quad u_i \geqq 0, \quad i = 1, \ldots, m. \tag{3.13}$$

3.3. LEMMA. Ha az $f(\mathbf{x})$. $-g_i(\mathbf{x})$, $i = 1, \ldots, m$ függvények kvázikonvexek az $\mathbf{x}^*$ pontban, $\nabla f(\mathbf{x}^*) \neq 0$, $\nabla g_i(\mathbf{x}^*) \neq 0$ $i \in B(\mathbf{x}^*)$ és az $\mathbf{x}^*$ pontban a *Kuhn—Tucker-féle feltételek* igazak, akkor az alábbi egyenlőtlenség is érvényes:

$$\mathbf{y}^T \nabla_{\mathbf{x}}^2 L(\mathbf{x}^*, \mathbf{u}^*)\mathbf{y} \geqq 0, \quad \text{ha} \quad \nabla g_i(\mathbf{x}^*)\mathbf{y} = 0, \quad i \in B(\mathbf{x}^*) = \{i | g_i(\mathbf{x}^*) = 0\}. \tag{3.14}$$

Bizonyítás. Tudjuk azt, hogy

$$\nabla_{\mathbf{x}}^2 L(\mathbf{x}^*, \mathbf{u}^*) = \nabla^2 f(\mathbf{x}^*) - \sum_{i=1}^{m} u_i^* \nabla^2 g_i(\mathbf{x}^*) = \nabla^2 f(\mathbf{x}^*) - \sum_{i \in B(\mathbf{x}^*)} u_i^* \nabla^2 g_i(\mathbf{x}^*).$$

Mivel az $\mathbf{x}^*$ pontban a $-g_i(\mathbf{x})$ függvények kvázikonvexek és $\nabla g_i(\mathbf{x}^*) \neq 0$ $i \in B(\mathbf{x}^*)$ így a 2.8. lemmát alkalmazva kapjuk, hogy

$$-\sum_{i \in B(\mathbf{x}^*)} u_i^* \mathbf{y}^T \nabla^2 g_i(\mathbf{x}^*)\mathbf{y} \geqq 0, \quad \text{ha} \quad \mathbf{y} \in S_3, \quad S_3 = \{\mathbf{y} | \nabla g_i(\mathbf{x}^*)\mathbf{y} = 0, \ i \in B(\mathbf{x}^*)\}.$$

Másrészt

$$\nabla f(\mathbf{x}^*) = \sum_{i \in B(\mathbf{x}^*)} u_i^* \nabla g_i(\mathbf{x}^*),$$

így $\nabla f(\mathbf{x}^*)\mathbf{y} = 0$, ha $\mathbf{y} \in S_3$.

Ha újra alkalmazzuk a 2.8. lemmát, akkor megkapjuk a (3.14) egyenlőtlenséget.

A (3.14) egyenlőtlenség nem más, mint az $\mathbf{x}^*$ pontban az optimalitás másodrendű szükséges feltétele.

3.4. *Megjegyzés.* Ha a 3.3. lemma feltételei teljesülnek és vagy a célfüggvényre vagy valamelyik aktív feltételre (ahol a megfelelő $u_i^* \neq 0$) az $\mathbf{x}^*$ pontban a 2.10. definíció értelmében vett szigorú pszeudokonvexitás igaz (a Γ halmazra nézve), akkor a (3.14) egyenlőtlenség élesen teljesül.

3.5. LEMMA. Ha a 3.2. tétel és a 3.3. lemma feltételei teljesülnek és az $\mathbf{x}^*$ pontban az $f(\mathbf{x})$, $-g_i(\mathbf{x})$, $i \in B(\mathbf{x}^*)$, $u_i^* \neq 0$ függvények közül legalább egy a 2.10 definíció értelmében szigorúan pszeudokonvex (a Γ halmazra nézve) akkor az $\mathbf{x}^*$ pont szigorú globális minimum pontja a (3.1) problémának.

Bizonyítás. A [9] 10.6.3. tételéből és az előbbiekből azonnal adódik az állítás.

4. A Kuhn—Tucker-féle feltételek ellenőrzése a SUMT-módszer esetén

Ebben a részben azzal a kérdéssel foglalkozunk, hogy hogyan tudjuk a SUMT-módszer alkalmazása esetén eldönteni,ha a büntetőfüggvények konvexitási tulajdonságait nem ismerjük, hogy a büntetőfüggvények lokális optimumainak egy sorozata mikor konvergál a feladat globális optimumához.

Először a logaritmikus büntetőfüggvényt vizsgáljuk.

Vezessük be az alábbi jelölést.

$$P(\mathbf{x}, r) = f(\mathbf{x}) - r \sum_{i=1}^{m} \log g_i(\mathbf{x}). \tag{4.1}$$

4.1. TÉTEL. Ha az $f(\mathbf{x})$, $g_i(\mathbf{x})$, $i=1, \ldots, m$ függvények kétszer folytonosan differenciálhatók és

a) az $f(\mathbf{x})$ az $\mathbf{x}^*$ pontban kvázikonvex, $\nabla f(\mathbf{x}^*) \neq \mathbf{0}$,

b) a Kuhn—Tucker-féle feltételek teljesülnek,

c) a $\nabla g_i(\mathbf{x}^*)$, $i \in B(\mathbf{x}^*)$ vektorok lineárisan függetlenek,

d) a szigorú komplementaritás teljesül,

e) az $\mathbf{x}^*$ pontban vagy az $f(\mathbf{x})$ vagy valamelyik $g_i(\mathbf{x})$, $i \in B(\mathbf{x}^*)$, $u_i^* \neq 0$ esetén a szigorú pszeudokonvexitás igaz, akkor az $r=0$ pontnak létezik olyan környezete, amelyhez egyértelműen hozzárendelhető egy folytonos, differenciálható függvénypár $(\mathbf{x}(r), \mathbf{u}(r))$, amelyek teljesítik a következő feltételeket:

$$\nabla f(\mathbf{x}(r)) - \sum_{i=1}^{m} u_i(r) \nabla g_i(\mathbf{x}(r)) = \mathbf{0}, \tag{4.2}$$

$$u_i(r) g_i(\mathbf{x}(r)) = r, \quad i = 1, \ldots, m. \tag{4.3}$$

Ebben a környezetben az $\mathbf{x}(r)$, $r>0$ görbe pontjait a megfelelő $P(\mathbf{x}, r)$ függvények szigorú lokális minimumai adják és

$$\mathbf{x}(r) \to \mathbf{x}^*, \quad \mathbf{u}(r) \to \mathbf{u}^*, \quad r \to 0 \quad \text{esetén.} \tag{4.4}$$

A tétel az [5]-ben található 14. tétel és a korábbiak alapján egyszerűen belátható. (Megjegyezzük, hogy az [5] 14. tételének a bizonyítása hibás, de kijavítható.)

Ha a (4.2), (4.3) egyenlőségekben az r értékével nullához tartva elvégezzük a határátmenetet, akkor éppen az $\mathbf{x}^*$ pontbeli *Kuhn—Tucker-féle feltételeket* kapjuk meg. Az $\mathbf{x}(r)$, $r>0$ görbe pontjait az optimum pont egy környezetében a SUMT-módszer segítségével algoritmikusan elő tudjuk állítani. Ezekben a pontokban automatikusan teljesül a (4.2) reláció, hiszen itt a megfelelő $P(\mathbf{x}, r)$ függvénynek szigorú lokális minimuma van.

A (4.3) relációból adódik, hogy a Lagrange-szorzók közelítései az optimum pont közelében az

$$\frac{r}{g_i(\mathbf{x}(r))}, \quad i = 1, \ldots, m \tag{4.5}$$

értékek. Ezért, ha a 4.1. tétel feltételei teljesülnek, az algoritmus folyamán a *Kuhn—Tucker-féle feltételek* ellenőrzésére elegendő csak az $\mathbf{x}(r)$, $r>0$, illetve a (4.5) értékek konvergenciáját vizsgálni.

Azonban a 4.1. tétel feltételei mellett a *Kuhn—Tucker-féle feltételek* teljesülése az $\mathbf{x}^*$ pont globális optimalitását jelenti.

Az algoritmus gyakorlati végrehajtása során véges lépésben keressük a feladat megoldását, ezért kihasználjuk azt, hogy az $\mathbf{x}(r)$, $\mathbf{u}(r)$, $r>0$ görbék folytonosak az optimum pont közelében. Ez azt jelenti, hogy ha az r értékét egy picit változtatjuk, akkor az $\mathbf{x}(r)$, $\mathbf{u}(r)$ értékek is keveset változnak, azaz véges lépés után is dönteni tudunk ezen értékek konvergenciájáról.

Mivel a *Kuhn—Tucker-féle feltételek* az optimalitás szükséges és elegendő feltételei, így ilyen módon csak optimumot találunk. Másrészt a SUMT-módszer alkalmazásakor ezek a feltételek egyszerűen ellenőrizhetők, hiszen minden érték rendelkezésünkre áll, így egy futó programba egyszerűen beépíthetők.

5. Globális optimumot adó SUMT algoritmus szigorúan pszeudokonvex függvények esetén

Ebben a részben algoritmust adunk a (3.1) feladat megoldására, ha a feladatban szereplő függvények a 2.10 definíció értelmében szigorúan pszeudokonvexek és az R kompakt halmaz. Az alkalmazott SUMT algoritmus GERENCSÉR [6] tételére épül és *Kuhn—Tucker pontot* keres.

5.1. TÉTEL Ha $f(\mathbf{x})$, $\mathbf{x}\in\Gamma$ egy szigorúan pszeudokonvex függvény a 2.10 definíció értelmében, és R a Γ halmaznak egy konvex, kompakt részhalmaza, $\nabla f(\mathbf{x})\neq\mathbf{0}$, $\mathbf{x}\in R$, akkor az $e^{cf(\mathbf{x})}$ függvény konvex lesz az R halmazon elég nagy c értékre.

Az itt közölt algoritmus azon az észrevételen alapszik, hogy ha *Kuhn—Tucker-féle pont*ot keresünk, akkor ezzel párhuzamosan ellenőrizhető az is, hogy az 5.1. tételben szereplő konstans értéke elég nagy-e.

Transzformáljuk a (3.1) feladatot az alábbi alakra

$$\begin{aligned} &\min e^{cf(\mathbf{x})}, \quad c > 0 \\ &h_i(\mathbf{x}) = 1 - e^{-cg_i(\mathbf{x})} \geqq 0, \quad i = 1, \ldots, m. \end{aligned} \tag{5.1}$$

Mivel szigorúan monoton transzformációt alkalmaztunk, ezért az $e^{cf(\mathbf{x})}$, $-h_i(\mathbf{x})$, $i=1, \ldots, m$ függvények is szigorúan pszeudokonvexek lesznek.

Másrészt a $h(\mathbf{x})_i$, $i=1, \ldots, m$ függvények ugyanazt a tartományt írják le, ezért ha az eredeti problémának $\mathbf{x}^*$ az optimum helye, akkor a transzformált feladatnak is az lesz. Könnyen megmutatható, hogy ha a 4.1. tétel feltételei teljesülnek az eredeti feladatra, akkor a transzformált feladatra is igazak.

Ezek alapján az (5.1) feladat esetén az algoritmust a következőképpen hajtjuk végre.

1. Megadunk egy induló c értéket, s a transzformált feladaton végrehajtjuk a SUMT-algoritmust. Ez azt jelenti, hogy megalkotjuk a

$$P(\mathbf{x}, r, c) = e^{cf(\mathbf{x})} - r \sum_{i=1}^{m} \log h_i(\mathbf{x})$$

függvényt, majd egy adott induló r_1 érték mellett ismert $\mathbf{x}_0 \in R_0$ pontból kiindulva minimalizáljuk a $P(\mathbf{x}, r_1, c)$ függvényt. Így megkapjuk egy $\mathbf{x}(r_1)$ lokális minimumát a $P(\mathbf{x}, r_1, c)$ függvénynek és a következő lépésben ebből a pontból kiindulva egy $r_2 < r_1$ érték mellett minimalizáljuk a $P(\mathbf{x}, r_2, c)$ függvényt. Ezt az eljárást addig folytatjuk, amíg pl. az

$$\|\mathbf{x}(r_{k+1}) - \mathbf{x}(r_k)\|, \quad \|e^{cf(\mathbf{x}(r_{k+1}))} - e^{cf(\mathbf{x}(r_k))}\|$$

különbségek egy-egy előre megadott érték alá nem süllyednek.

2. Megállapítjuk a 3.2. tétel alapján, hogy optimumot találtunk-e. Ezt a következő kritériumok alapján dönthetjük el:

a) $\nabla f(\mathbf{x}^*) \neq \mathbf{0}$, $\nabla h_i(\mathbf{x}^*) \neq \mathbf{0}$, $i \in B(\mathbf{x}^*)$

b) ha az $\mathbf{x}(r_k)$ értékek keveset változnak, akkor a (4.5) képlettel megadott *Lagrange-szorzók* közelítései is keveset változnak.

3. Ha az algoritmussal optimumot találtunk, akkor meghatározzuk az eredeti feladat optimum értékét és befejezzük a számolást. Ha nem, akkor megnöveljük a c értékét és a transzformált feladaton újra végrehajtjuk az eljárást.

Mivel a c értékének a növelésével véges lépésben konvex programozási problémát kapunk, így az algoritmussal biztosan globális optimumot határozunk meg. Elképzelhető azonban, hogy sok probléma esetében még nem konvex programozási feladatnak találjuk meg így a globális optimumát.

Megjegyezzük, hogy ezt az algoritmust hasonlóan fel lehet építeni az „$1/x$" belső pontos és a „$\min\{0, x\}^2$" külső pontos büntető függvényekre is.

IRODALOM

[1] ARROW, K. J. and ENTHOVEN, A. C., "Quasi-concave programming", *Econometrica* **29** (1961) 778—800.

[2] Avriel, M., "r-convex functions," *Mathematical Programming* **2** (1972) 309—323.

[3] Cottle, W., and Ferland, A., "On pseudo-convex functions of nonnegative variables," *Mathematical Programming* **1** (1971) 95—101

[4] FERLAND, J. A., "Mathematical programming problems with quasi-convex objective functions", *Mathematical Programming* **3** (1972) 296—301.

[5] FIACCO, A. V. and MCCORMICK, G. P., *Nonlinear Programming: Sequential Unconstrained Minimization Techniques* (Wiley and Sons, New York, 1968).

[6] GERENCSÉR, L., "On a close relation between quasiconvex and convex functions and related investigations", *Math. Operationsforschung und Statistik* **4** (1973) 201—211.

[7] GERENCSÉR, L., „Nemlineáris programozási feladatok megoldása szekvenciális módszerekkel", *MTA SZTAKI Tanulmányok* 49/1976.

[8] KÉRI, G., "An examination of nonnegativity and quasiconvexity conditions of quadratic form on the non-negativ orthant", *Studia Scientiarum Mathematicarum Hungarica* **6** (1971) 193—196.
[9] Luenberger, D. G., *Introduction to linear and nonlinear programming* (Addison-Wesley Publisting Company, Inc. 1973).
[10] MANGASARIAN, O. L., "Pseudo-convex functions", *SIAM Journal on Control* **3** (1965) 281—290.
[11] MANGASARIAN, O. L., *Nonlinear Programming* (McGraw-Hill Book Company, 1969).
[12] MARTOS, B., "The direct power of adjacent vertex programming methods", *Management Science* **12** (1965) 241—252.
[13] Martos, B., *Nonlinear programming theory and methods*, (Akadémiai Kiadó, Bp., 1975).
[14] PRÉKOPA, A., „Sztohasztikus rendszerek optimalizálási problémáiról", doktori értekezés. Magyar Tudományos Akadémia, Budapest, 1970.
[15] PRÉKOPA, A., „Eine Erweiterung der sogenanter Methode der ‚zulassige Richtungen' der nichtlinearen Optimierung auf der Fall quasikonkaver Restriktionsfunktionen", *Math. Operationsforschung und Statistik*, **4** (1973).
[16] ZANGWILL, W. I., *Nonlinear Programming: A Unified Approach* (Inc. Englewood Cliffs, N. J., 1969).

(Beérkezett: 1975. október 1.)

(Újra beérkezett: 1977. február 10.)

RAPCSÁK TAMÁS
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST I., ÚRI U. 49.

THE SUMT METHOD FOR SOLVING NON CONVEX PROGRAMMING PROBLEMS

T. RAPCSÁK

This paper deals with the problem how we can come to a conclusion that local optima of the penalty functions converge to the global optimum of a mathematical programming problem. First we examine the properties of the quasiconvex and pseudoconvex functions, the first and second order necessary and sufficient conditions for the global optimum. Thus applying the SUMT method and seeking Kuhn—Tucker point, we observe the global optimality. Finally it is given a SUMT algorithm to search the global optimum of a problem having generalized convex functions.

INDEX

1828—1978

MEGJELENT AZ AKADÉMIAI KÖNYVKIADÁS 150. ÉVÉBEN

A kiadásért felel az Akadémiai Kiadó igazgatója
Műszaki szerkesztő: Sándor István
A kézirat nyomdába érkezett: 1977. XI. 30. Terjedelem: 25,2 (A/5) ív
77-5242 — Szegedi Nyomda — Felelős vezető: Dobó József

ÚTMUTATÁS A SZERZŐKNEK

Az Alkalmazott Matematikai Lapok csak magyar nyelvű dolgozatokat közöl. A kéziratok gépelését olyan formában kérjük, hogy minden gépelt oldal 25, egyenként átlag 50 betűhelyes sort tartalmazzon. A közlésre szánt dolgozatokat három példányban a felelős szerkesztő címére kell beküldeni:

Prékopa András, felelős szerkesztő, MTA SZTAKI
1502 Budapest XI., Kende u. 13—17.

A kéziratok szerkezeti felépítésének a következő követelményeket kell kielégíteni. A fejlécnek tartalmaznia kell a dolgozat címét, a szerző teljes nevét, valamint annak a városnak a nevét, ahol a szerző dolgozik. A fejléc után egy, képletet nem tartalmazó, legfeljebb 200 szóból álló kivonatot kell minden esetben megadni. A dolgozatot címmel ellátott szakaszokra kell bontani, és az egyes szakaszokat arab sorszámmal kell ellátni. Az esetleges bevezetésnek mindig az első szakaszt kell alkotnia. Az irodalomjegyzék mindig az utolsó szakasz kell hogy legyen, és azt nem kell sorszámmal ellátni. Az irodalomjegyzék után, a kézirat befejezéseképpen fel kell tüntetni a szerző teljes nevét és a munkahelye (illetve lakása) pontos postai címét. A dolgozatban előforduló képleteket szakaszonként újrakezdődően, a képlet előtt két zárójel közé írt kettős számozással kell azonosítani. Természetesen nem szükséges minden képletet számozással ellátni. Az esetleges definíciókat és tételeket (segédtételeket és lemmákat) ugyancsak szakaszonként újrakezdődő, kettős számozással kell ellátni. Kérjük a szerzőket, hogy ezeket, valamint a tételek bizonyítását a szövegben kellő módon emeljék ki. Minden dolgozathoz csatolni kell egy angol, német, francia vagy orosz nyelvű, külön oldalra gépelt összefoglalót. Amennyiben lehetséges, kérjük a nyomtatás számára különösen nehézkes matematikai jelölések használatának az elkerülését.

A dolgozat ábráit és az esetleges lábjegyzeteket a dolgozat végén, különálló lapokon kérjük beküldeni. Mind az ábrákat, mind a lábjegyzeteket a dolgozat szakaszokra bontásától független, folytatólagos arab sorszámozással kell ellátni. Az ábrák elhelyezését a dolgozat megfelelő helyén, széljegyzetként feltüntetett, ábraazonosító sorszámokkal kell megadni. A lábjegyzetekre a dolgozaton belül az azonosító sorszám felső indexként használatával lehet hivatkozni.

Az irodalmi hivatkozások formája a következő. Minden hivatkozást fel kell sorolni a dolgozat végén található irodalomjegyzékben, a szerzők, illetve társszerzők esetén az első szerző neve szerinti alfabetikus sorrendben úgy, hogy külön, de folytatólagos sorszámozású listát alkossanak a latin és a cirill betűs nevű szerzők műveire vonatkozó hivatkozások, és mindkét részben a megfelelő alfabetikus sorrend legyen kialakítva. A folyóiratban megjelent cikkekre [1], a könyvekre [5], a kötetben megjelent dolgozatokra [4], a disszertációkra [3] és a gépi program leírásokra [2] a következő minta szerint kell hivatkozni:

[1] Farkas, J., »Über die Theorie der einfachen Ungelichungen«, *Journal für die reine und angewandte Mathematik* **124** (1902) 1—27.

[2] Kéri, G., „DUALSIMP", rutin a CDC 3300-as gépekre (Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézete, CDC 3300 felhasználói ismertetők 2. 1973. május) 19—20.

[3] Prékopa, A., „Sztohasztikus rendszerek optimalizálási problémáiról", doktori értekezés. Magyar Tudományos Akadémia, Budapest, 1970.

[4] Prabhu, N. U., "Recent research on the ruin problem of collective risk theory", in: *Inventory Control and Water Storage* Ed. A. Prékopa (János Bolyai Mathematical Society and North-Holland Publishing Company, Amsterdam—London, 1973) 221—228.

[5] Zoutendijk, G., *Methods of Feasible Directions* (Elsevier Publishing Company, Amsterdam and New York, 1960).

A dolgozatok szövegében az irodalmi hivatkozás számait szögletes zárójelben kell megadni, mint például [5] vagy [4, 76—78]. A szerzők a dolgozatukról 100 darab különlenyomatot kapnak, ezek költsége — nyomott oldalanként 25 forint — a szerzői díjat terheli.

Ára: 54,— Ft

INDEX: 26498
ISSN 0133—3399

TARTALOMJEGYZÉK