Alkalmazott matematikai lapok

1980/1-2

A MAGYAR TUDOMÁNYOS AKADÉMIA
MATEMATIKAI ÉS FIZIKAI TUDOMÁNYOK
OSZTÁLYÁNAK KÖZLEMÉNYEI

6.

KÖTET

AKADÉMIAI KIADÓ, BUDAPEST

A MAGYAR TUDOMÁNYOS AKADÉMIA
MATEMATIKAI ÉS FIZIKAI TUDOMÁNYOK OSZTÁLYÁNAK
ALKALMAZOTT MATEMATIKAI LAPJA

VI. 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, főszerkesztő
1502 Budapest, 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 100 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 6 (1980) 1—16

A BÁZISBÓL KILÉPŐ VEKTOR MEGHATÁROZÁSÁNAK EGY MÓDJA A SZIMPLEX MÓDSZER ELSŐ FÁZISÁBAN

MAROS ISTVÁN

Budapest

A szimplex módszer első fázisa a lineáris programozási (LP) feladatok egy lehetséges (fizibilis) megoldásának megkeresésére szolgál, de használható lineáris feltétellel rendelkező egyéb feladatok esetén is hasonló célra, sőt ilyen feltételrendszerek konzisztenciájának a megállapítására is. LP feladatok esetén a klasszikus első fázisú eljárások az alábbi feltételek mellett működnek:

(i) A bázisba belépő változó fizibilis értéket vesz fel.

(ii) A bázisban levő fizibilis változók a transzformáció után is fizibilis értéken maradnak.

(iii) A bázisból kilépő vektor fizibilis értéken hagyja el a bázist (alsó vagy felső korláton).

Az (ii) feltétel feloldásával új lehetőségek kínálják magukat a szimplex módszer keretein belül. A cikk is egy ehhez kapcsolódó eljárást mutat be, amitől — pusztán a bázisból kilépő változó másként történő meghatározása által — hatékonyabb iterációs lépések várhatók az első fázisban, különösen degenerált bázisok esetén.

1. Bevezetés

A szimplex módszer első fázisa közismerten a lineáris programozási (LP) feladatok egy lehetséges (fizibilis) megoldásának a megkeresésére szolgál, sőt a lineáris feltételrendszerrel rendelkező egyéb feladatok esetén is használják fizibilis pontok előállítására, illetve a feltételrendszer konzisztens/inkonzisztens voltának a kimutatására.

A lineáris programozásban az első fázist tulajdonképpen „szükséges rossz"-nak szokták tekinteni, hiszen ekkor az LP feladat eredeti célfüggvénye egyáltalán nem, vagy csak igen korlátozottan érvényesül. Ennek a résznek a lerövidítése tehát az LP problémák megoldásának hatékonysága szempontjából fontos (gyakorlati) feladat. Már DANTZIG is a szimplex módszert hívta segítségül fizibilis megoldás előállítására és azóta is ennek különféle variánsai használatosak. Az ilyen irányú fejlesztések mind a bázisba belépő, mind pedig a bázist elhagyó változó meghatározásának mikéntjére kiterjednek (bizonyos esetekben ezek egymással erősen összefüggnek), hogy csak az algoritmus jellegű fejlesztéseket említsük, nem beszélve a stabilabb numerikus viselkedést célzó eljárások kidolgozásáról.

A klasszikus első fázisú eljárások (pl. [1], [5]) általában az alábbi feltételek mellett működnek:

(i) a bázisba belépő változó fizibilis értéket vesz fel;

(ii) a bázisban levő fizibilis változók a transzformáció után is fizibilis értéken maradnak;

(iii) a bázisból kilépő vektor fizibilis értéken hagyja el a bázist (alsó vagy felső korláton).

Ezekből rögtön következik, hogy ha induláskor a bázison kívüli változók fizibilis értéken vannak, akkor ez végig igaz lesz a mindenkori bázison kívüliekre.

Az (i)—(iii) feltételek még nem határozzák meg egyértelműen egy adott belépő vektor esetén a kilépő vektort. Ezt a lehetőséget kihasználva különféle eljárások készültek, melyek mind azt célozták, hogy az első fázis minél hamarabb befejeződjék. Az (ii) feltétel feloldásával új lehetőségek kínálják magukat, amelyektől — pusztán a kilépő vektor másként történő meghatározása által — még hatékonyabb iterációs lépések várhatók. Ezzel kapcsolatos ötletet említett TOMLIN is a Budapesten 1976-ban tartott *IX. Matematikai Programozási Szimpozionon.* A továbbiakban egy ilyen elgondolásnak a kidolgozásáról lesz szó, de utalás történik arra is, hogyan tudja szolgálni a degenerációs ciklizálás elkerülését és a nagyobb numerikus stabilitást az eljárás.

Kimutatjuk, hogy tulajdonképpen az (i)—(iii) feltételek által meghatározott keretek általánosításáról van szó.

Végül beszámolunk számítástechnikai tepasztalatokról, melyben egyúttal bemutatunk összehasonlító vizsgálati eredményeket egy, az (i)—(iii) feltételeket teljesítő, elég alaposan kidolgozott eljárással szemben.

2. A megoldandó LP feladat

A következőkben az alábbi LP feladatról lesz szó:

$$y_i + \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, \ldots, m \tag{2.1}$$

és az egyes változók az alábbi csoportok valamelyikébe esnek:

(2.2)

Típus	Fizibilis értéktartomány		Megjegyzés
0	$y_i = 0$	$x_j = 0$	
1	$0 \leqq y_i \leqq v_i$	$0 \leqq x_j \leqq u_j$	v_i, u_j véges
2	$0 \leqq y_i \leqq +\infty$	$0 \leqq x_j \leqq +\infty$	
3	$-\infty \leqq y_i \leqq +\infty$	$-\infty \leqq x_j \leqq +\infty$	

A (2.1) alatti feltételrendszer bármelyik sorát lehet célfüggvénynek tekinteni. Az egyöntetűbb tárgyalás kedvéért feltesszük, hogy a célfüggvényhez tartozó y_i változót maximalizáljuk.

Belátható (pl. [4]), hogy a (gyakorlatban felmerülő) legáltalánosabb alakú LP feladatok a fenti alakra hozhatók egyszerű transzformációk segítségével. Számítógépes megoldás esetén ezek a transzformációk az input során könnyen elvégezhetők, ugyanakkor az implementált LP optimalizáló algoritmusok döntő többsége a (2.1)—(2.2)-vel felírt alakot használja csakúgy, mint a tárgyalásban később ismertetendő LIPROS LP programcsomag is.

A (2.1)-ben szereplő x_j változókat *strukturális változók*nak, míg az y_i-ket *logikai változók*nak nevezzük. A továbbiakban feltesszük, hogy a logikai változókat 1-től m-ig, a strukturálisakat $m+1$-től $m+n$-ig számoztuk és a felső korlátokat egységesen u_j-vel jelöljük.

Jelöljük $\boldsymbol{\beta}$-val az aktuális bázismegoldást, vagyis

$$\boldsymbol{\beta} = \mathbf{B}^{-1}\bar{\mathbf{b}}, \tag{2.3}$$

ahol $\mathbf{B}^{-1}$ a bázisinverz és $\bar{\mathbf{b}}$ az eredeti $\mathbf{b}$ jobb oldal vektorból keletkezik oly módon, hogy a bázison kívül felső korláton levő változóknak (ezek indexhalmaza: J) megfelelő oszlopok hatását figyelembe vesszük:

$$\bar{\mathbf{b}} = \mathbf{b} - \sum_{j \in J} u_j \mathbf{a}_j. \tag{2.4}$$

További jelölések:

I_B a bázisváltozók indexhalmaza,
I_0 a 0 típusú bázisváltozók indexhalmaza,
I_1 az 1 típusú bázisváltozók indexhalmaza,
I_2 a 2 típusú bázisváltozók indexhalmaza,
I_3 a 3 típusú bázisváltozók indexhalmaza.

Nyilvánvalóan fennáll az

$$I_B = I_0 \cup I_1 \cup I_2 \cup I_3 \tag{2.5}$$

összefüggés.

Említettük, hogy a lineáris programozás első fázisa egy fizibilis (megengedett) megoldás megkeresésére szolgál, továbbá infizibilis értéken levő változók csak bázisváltozók lehetnek. A bázisváltozókat ennek megfelelően feloszthatjuk fizibilitás szempontjából. Az egységes tárgyalás érdekében a 0 típusú változókra is értelmezzük a felső korlátot és ennek értékét értelemszerűen 0-nak tekintjük. Ezek előrebocsátásával a bázisváltozók fizibilitási állapotát kifejező három indexhalmazt definiálunk:

$$M = \{i:\ i \notin I_3 \wedge \beta_i < 0\},$$

$$P = \{i:\ i \in I_0 \cup I_1 \wedge \beta_i > u_i\},$$

$$F = I_B \setminus (M \cup P).$$

M tehát a mínusz irányban, P a plusz irányban infizibilis változók indexhalmaza, F pedig a fizibilis értéken levő változóké. Egy 3-as típusú bázisváltozó mindig az F halmazhoz tartozik.

Egy bázismegoldás infizibilitásának mértékét a következőképpen definiáljuk:

$$w = \sum_{i \in M} \beta_i - \sum_{i \in P} (\beta_i - u_i). \tag{2.6}$$

Ezt úgy értelmezhetjük, hogy w az infizibilis bázisváltozók fizibilitási tartománytól való távolságának a -1-szerese. Megjegyezzük, hogy ez a definíció érdemben különbözik az Orchard—Hays [4] által használt

$$w = \sum_{i \in M} \beta_i - \sum_{i \in P} \beta_i \tag{2.7}$$

mértéktől, amint azt a későbbiekben kimutatjuk. Nyilvánvaló, hogy $w \leqq 0$, és a 0-t éppen akkor éri el, amikor a definícióban szereplő két halmaz üressé válik, azaz a bázismegoldás fizibilis. Az első fázisban a cél tehát egy olyan megoldás megtalálása, melyre $w=0$ teljesül. Ez elérhető azáltal, hogy maximalizáljuk w-t a (2.1) feltételek

mellett. Ezt nevezzük W feladatnak. Ha elértük a $w=0$ elméleti maximumot, akkor egyben találtunk egy fizibilis megoldást a (2.1)—(2.2) feltételrendszerhez.

A W feladat tulajdonképpen nem egy szokásos értelemben vett LP feladat. Ahogy az M és P halmazok változnak (az (ii) feltétel esetén ez csökkenést jelent), úgy változik maga a célfüggvény összetétele is.

3. A W feladat megoldásáról

A W feladat megoldására a szimplex módszer alapvető technikáját lehet alkalmazni. Tegyük fel, hogy egy x_j bázison kívüli változó nulla szinten van és azt vizsgáljuk meg, hogy ennek a változónak a növelése milyen hatással van w-re.

Paraméterezzük x_j-nek a nulláról való elmozdulását t-vel. Ekkor a (2.1) egyenlőségek fennállásához a bázisváltozók értéke is megváltozik t függvényében, amit az $\mathbf{f}(t)$ vektorral jelölünk. Így a

$$\mathbf{B}\mathbf{f}(t)+t\mathbf{a}_j = \bar{\mathbf{b}} \tag{3.1}$$

egyenlőséget kapjuk (2.1)-ből. Jelöljük α_j-vel az $\mathbf{a}_j$ vektor $\mathbf{B}$-beli képét: $\alpha_j=\mathbf{B}^{-1}\mathbf{a}_j$. Ennek segítségével (3.1)-ből azonnal megkaphatjuk a bázisváltozók értékét t függvényében:

$$\mathbf{f}(t) = \boldsymbol{\beta}-t\boldsymbol{\alpha}_j. \tag{3.2}$$

Itt az egyszerűbb jelölés érdekében nem tüntetjük fel külön a bal oldalnak a j-től való függését.

Tegyük fel, hogy egy olyan $\mathbf{B}$ bázisnál vagyunk, melyre $w<0$. Erre igaz a következő:

3.1. LEMMA. Ahhoz, hogy x_j 0 szinten bázison kívüli változó értékének növelése javítani tudja w-t, szükséges, hogy teljesüljön rá a

$$d_j = \sum_{i\in M} \alpha_{ij} - \sum_{i\in P} \alpha_{ij} < 0 \tag{3.3}$$

egyenlőtlenség, ahol az α_{ij} értékek az α_j vektor komponensei.

Ennek az állításnak az igazolására elég megnézni, hogy w hogyan változik t növelésekor. Tegyük fel, hogy $t>0$ elég kicsi ahhoz, hogy az M és P halmazok változatlanok maradjanak. Ekkor w megváltozása:

$$\begin{aligned}\Delta w &= \sum_{i\in M}(f_i(t)-f_i(0)) - \sum_{i\in P}((f_i(t)-u_i)-(f_i(0)-u_i)) = \\ &= \sum_{i\in M}(-t\alpha_{ij}) - \sum_{i\in P}(-t\alpha_{ij}) = \\ &= -t\Big(\sum_{i\in M}\alpha_{ij} - \sum_{i\in P}\alpha_{ij}\Big) = -td_j.\end{aligned}$$

Innen $\Delta w\geqq 0$-hoz $d_j<0$ a triviális követelmény.

Ha az M és P halmazok változatlansága csak $t=0$-ra teljesül, akkor az azt jelenti, hogy a bázis degenerált és az x_j változó csak 0 szinten tudna belépni a bázisba az (ii) feltétel miatt.

Ha bázison kívüli változót negatív irányban mozdítunk el (1-es típusú változó felső korlátról, vagy 3-as típusú negatív értékkel jön be), akkor értelemszerűen $\Delta w \geqq 0$-hoz $d_j > 0$ szükséges.

(3.3)-at valójában w x_j szerint vett parciális deriváltjának lehet tekinteni.

A W feladat megoldásának egyik lépése tehát abból áll, hogy megvizsgáljuk d_j értékét minden x_j bázison kívüli változóra, melynek típusa 0-tól különböző és eldöntjük, hogy valamilyen irányban elmozdítva javíthat-e az infizibilitások w mértékén. Minthogy a megfelelők közül még további szempontok szerint célszerű választani, ez a kiválasztás igen gondos mérlegelést, több tényező együttes és esetleg dinamikus figyelembevételét igényli. Ennek a részleteiről azonban e helyen nem kívánunk többet mondani és egyelőre feltesszük, hogy valamilyen módon meghatároztunk egy potenciális javító vektort. A szokásos eljárásban a t lépéshossz és a kilépő változó meghatározása ezután már az (ii), (iii) feltételek alapján történik. A továbbiakban azt vizsgáljuk, hogy az (ii) feltétel relaxálása esetén a belépő változó melyik bázisváltozóval cseréljen helyet és milyen t értéknél a w mérték minél nagyobb mérvű javítása érdekében.

4. A bázisváltozók mozgása

A bázisváltozók mozgásának vizsgálatánál alapvető célunk az, hogy a belépő változó nagyságának függvényében a (3.1) egyenlőségek teljesülése mellett meghatározzuk, hogy hogyan alakul az egyes bázisváltozók fizibilitási állapota, vagyis milyen értékkel járulnak hozzá w-hez. Ezt a vizsgálatot a (3.2)-ben felírt $\mathbf{f}(t)$ függvényre komponensenként végezzük el. A j indexet a jobb oldalról is elhagyjuk és az

$$f_i(t) = \beta_i - t\alpha_i, \quad i = 1, \ldots, m \tag{4.1}$$

függvényeket mint a bázisváltozók t-től függő értékeit tekintjük. Alapesetben, a pozitív irányú elmozdulásnál a $t \geqq 0$ értékekről van szó. A továbbiakban elegendő csak a nem 3-as típusú bázisváltozókat nézni, ugyanis a 3-as típusúak minden értéke fizibilis, így az infizibilitás szempontjából nem játszanak szerepet. A nem 3-as típusú bázisváltozók index-halmazát I-vel jelöljük:

$$I = I_B \setminus I_3.$$

Egy I-beli bázisváltozó negatív irányban infizibilis, ha értéke negatív és egy $I_0 \cup I_1$-beli pozitív irányban infizibilis, ha értéke nagyobb a felső korlátjánál.

Felhasználva a következő jelölést:

$$Z^- = \begin{cases} 0, & \text{ha} \quad Z \geqq 0, \\ Z, & \text{ha} \quad Z < 0, \end{cases}$$

$$Z^+ = \begin{cases} Z, & \text{ha} \quad Z > 0, \\ 0, & \text{ha} \quad Z \leqq 0, \end{cases}$$

az infizibilitás mértéke t függvényében az alábbi alakban írható fel:

$$w(t) = \sum_{i\in I} (f_i(t))^- - \sum_{i\in I_0\cup I_1} (f_i(t)-u_i)^+ = \tag{4.2}$$

$$= \sum_{i\in I} (\beta_i - t\alpha_i)^- - \sum_{i\in I_0\cup I_1} (\beta_i - t\alpha_i - u_i)^+.$$

Ebből az alakból rögtön látható, hogy $w(t)$ egy folytonos lineáris törtvonalfüggvény. Töréspontja ott van, ahol valamelyik változó fizibilitási állapotában változás történik. Az is nyilvánvaló, hogy a (2.6)-ban definiált w értéket éppen a $w(0)$ szolgáltatja. Ilyen értelemben $w(t)$-t w függvényszerű kiterjesztésének lehet tekinteni. Ha erre a célra a [4]-ben szereplő és (2.7)-ben idézett w-t használnánk, akkor egy nem folytonos függvényt kapnánk (ugyanis az $i\in I_0\cup I_1$ változók hozzájárulása a második szummához egy u_i értékű ugrást jelentene a korlát átlépésekor), így ez nem lenne alkalmas a további vizsgálatok elvégzésére.

(4.2)-ben egy $f_i(t)$-nek mindaddig van nullától különböző hozzájárulása az első szummához, amíg t olyan, hogy $f_i(t)<0$, vagyis $i\in M$. Hasonlóképpen $i\in I_0\cup I_1$ esetén $f_i(t)-u_i$ addig járul hozzá a második szummához, amíg $i\in P$. Egy bázisváltozó fizibilitási állapotában akkor következhet be változás, amikor eléri fizibilitási tartományának a határát. T_a-val jelöljük azt a t értéket, melyre az i-edik bázisváltozó eléri alsó korlátját (vagyis a 0-t).

$$T_a = \beta_i/\alpha_i, \quad \alpha_i \neq 0. \tag{4.3}$$

Az egyedi felső korláttal rendelkező változók esetén T_f-fel jelöljük a felső korlát elérésének helyét:

$$T_f = (\beta_i - u_i)/\alpha_i, \quad \alpha_i \neq 0. \tag{4.4}$$

Az $\alpha_i=0$ eset jelen szempontból érdektelen, mivel egy ilyen i-hez tartozó bázisváltozó nem mozdul el, így fizibilitási állapota sem változik.

A 0 típusú változók esetén $u_i=0$ miatt $T_a=T_f$ adódik, ezt azonban formálisan két értéknek tekintjük.

Tekintettel arra, hogy a $w(t)$ függvényt a $t\geqq 0$ tartományban vizsgáljuk, ezért nyilvánvaló, hogy $w(t)$ töréspontjai azon T_a és T_f értékek közül kerülnek ki, melyek nem negatívok, és annyi töréspont lesz, ahány különböző T_a és T_f érték adódik. Rendezzük az összes (tehát nemcsak a különböző) T_a, T_f értéket nagyság szerint növekvő sorrendbe és alkalmazzuk az alábbi jelölést:

$$0 \leqq t_1 \leqq t_2 \leqq \ldots \leqq t_K, \tag{4.5}$$

ahol K az értékek száma.

5. A $w(t)$ függvény vizsgálata

A $w(t)$ függvény vizsgálatához először azt nézzük meg, hogyan változik az i-edik bázisváltozó fizibilitása t függvényében. Tekintettel arra, hogy

$$f_i(t) = \beta_i - t\alpha_i$$

ezért a fizibilitás az i-edik bázisváltozó típusának, a β_i fizibilitásának és α_i előjelének a függvénye. Elsődlegesen α_i szerint teszünk különbséget.

I. Eset. $\alpha_i<0$.

Ebben az esetben $f_i(t)$ növekvő függvény $-\alpha_i$ meredekséggel.

1) $\beta_i<0$, tehát $t=0$-ra $i\in M$. Ekkor a bázisváltozó, típusától függetlenül ($i\in I$) a $T_a=\beta_i/\alpha_i>0$ pontban fizibilis lesz, vagyis i kikerül az M halmazból és bekerül F-be. Ugyanekkor megszűnik $f_i(t)$ hozzájárulása $w(t)$ első szummájához. Ha t értékét tovább növeljük $i\in I_2$ esetén $f_i(t)$ végig fizibilis marad, $i\in I_0\cup I_1$ esetén pedig a $T_f=(\beta_i-u_i)/\alpha_i$ pontban eléri felső korlátját és attól kezdve $-\alpha_i$ mértékkel járul hozzá $w(t)$ második szummájához és i az F halmazból átkerül P-be.

2) $\beta_i>u_i$. Ez csak $i\in I_0\cup I_1$ esetben fordulhat elő, és ekkor $t=0$-ra $i\in P$. Miután $f_i(t)$ most növekvő függvény és már $t=0$-ra nagyobb, mint u_i, ezért minden $t>0$-ra $i\in P$ marad.

3) β_i fizibilis, tehát $t=0$-ra $i\in F\cap I$. Ha $i\in F\cap I_2$, akkor $f_i(t)$ minden $t>0$-ra is fizibilis marad. Ha $i\in F\cap(I_0\cup I_1)$, akkor $f_i(t)$ a $T_f=(\beta_i-u_i)/\alpha_i$ pontban eléri felső korlátját és attól kezdve $-\alpha_i$ mértékben járul hozzá $w(t)$ második szummájához és i az F halmazból átkerül a P-be.

II. Eset. $\alpha_i>0$.

Ebben az esetben $f_i(t)$ csökkenő függvény $-\alpha_i$ meredekséggel.

1) $\beta_i<0$, tehát $t=0$-ra $i\in M$ minden $i\in I$-re. Ekkor a $t>0$ értékre $f_i(t)<\beta_i$, így végig $i\in M$ marad.

2) $\beta_i>u_i$. Ez csak $i\in I_0\cup I_1$ esetben fordulhat elő és ekkor $t=0$-ra $i\in P$. Növekvő t értékekre a $T_f=(\beta_i-u_i)/\alpha_i$ pontban $f_i(t)$ fizibilis lesz, megszűnik hozzájárulása $w(t)$ második szummájához és i átkerül P-ből F-be. Tovább növelve t értékét a $T_a=\beta_i/\alpha_i$ ponttól kezdve i átkerül F-ből M-be és $f_i(t)-\alpha_i$ mértékkel járul hozzá $w(t)$ első szummájához.

3) β_i fizibilis, tehát $t=0$-ra $i\in F\cap I$. Ekkor a $T_a=\beta_i/\alpha_i$ pontig $i\in F$ marad, onnantól pedig i átkerül F-ből M-be és $f_i(t)-\alpha_i$ mértékkel járul hozzá $w(t)$ első szummájához.

Ezek után rátérhetünk $w(t)$ vizsgálatára. Feltételezésünk értelmében olyan bázisnál vagyunk, amelyre $w(0)<0$. Ekkor t értékét növelve a $[0, t_1]$ intervallumban a $w(t)$ függvény

$$r_1=-\sum_{i\in M}\alpha_i-\sum_{i\in P}\alpha_i \tag{5.1}$$

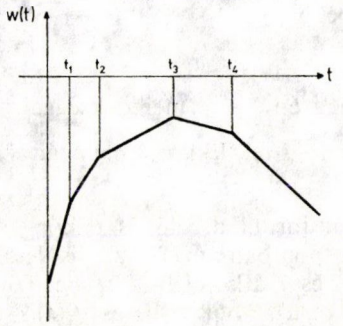
meredekséggel, a t_1 pontig összesen $r_1t_1\geqq 0$ értékkel növekszik, hiszen r_1 a belépő x_j változóhoz tartozó, a 3.1 lemmában szereplő $-d_j$-vel azonos. A $t=t_1$ pontban valamelyik (esetleg több) bázisváltozó fizibilitási állapota megváltozik. Legyen ez az i_1-edik bázisváltozó. Ha $\alpha_{i_1}<0$, akkor $f_{i_1}(t)$ növekedve érte el valamelyik határát, így az i_1 index vagy az M halmazból került át F-be, vagy (ha $i_1\in(I_0\cup I_1)\cap F$ teljesül) F-ből P-be. Mindkét esetben azonban a t_1 ponttól $w(t)$ meredeksége $r_2=r_1+\alpha_{i_1}$ lesz, amint ez (5.1) alapján látható. Figyelembe véve, hogy $\alpha_{i_1}<0$ ez azt jelenti, hogy r_1 csökkent, mégpedig $|\alpha_{i_1}|$ értékkel, tehát

$r_2 = r_1 - |\alpha_{i_1}|$. Ha $\alpha_{i_1} > 0$, akkor $f_{i_1}(t)$ csökkenve érte el valamelyik határát, így az i_1 index vagy a P halmazból került F-be, vagy F-ből M-be. Mindkét esetben a t_1 ponttól $w(t)$ meredeksége — szintén (5.1) alapján — $r_2 = r_1 - \alpha_{i_1}$ lesz. Miután most $\alpha_{i_1} > 0$, ezért ebben az esetben is igaz, hogy $r_2 = r_1 - |\alpha_{i_1}|$, vagyis meredekség mindkét esetben csökken. Tekintettel arra, hogy a fenti gondolatmenet minden t_k, $k = 1, \ldots, K$ pontra elmondható, ezért igaz az, hogy $w(t)$ meredeksége minden t_k töréspontban α_{i_k} értékkel csökken: $r_{k+1} = r_k - |\alpha_{i_k}|$. Ezzel viszont beláttuk az alábbi fontos tételt:

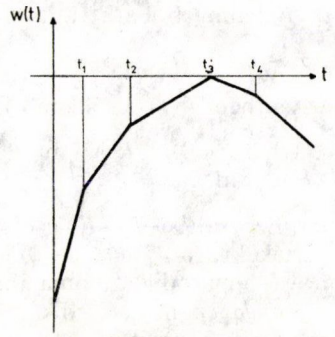
5.1. TÉTEL. $w(t)$ egy konkáv törtvonalfüggvény, melynek töréspontjai a (4.5)-ben definiált t_k, $k = 1, \ldots, K$ értékek, és $k+1$-edik szakaszának meredeksége: $r_{k+1} = r_k - |\alpha_{i_k}|$. Egy szakasz hossza egybeeső pontok esetén nulla is lehet.

Megjegyzendő, hogy a bizonyítás során sehol sem használtuk ki, hogy a t_k értékek különbözőek. Ez egyben azt is jelenti, hogy egybeeső pontok esetén a következő pozitív hosszúságú szakasz meredekségének a csökkenése az előző, ilyenhez képest a pontokhoz tartozó $|\alpha_{i_k}|$ értékek összegével lesz egyenlő.

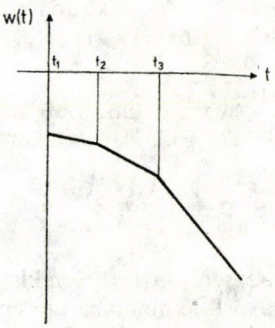
Az alábbiakban bemutatjuk $w(t)$ néhány lehetséges alakját:



1. ábra



2. ábra



3. ábra

Az 1. ábra egy átlagos típusú alakot szemléltet. A 2. ábrán $w(t)$ eléri elméleti maximumát, a 0-t. A 3. ábra érdekes jelenséget mutat. A t_1 pont a 0-ba esik (esetleg több is eshet egyszerre ide) és a hozzá tartozó α_{i_1} érték olyan, hogy azzal már $r_2 = r_1 - |\alpha_{i_1}| < 0$ lesz. Ez azt jelenti, hogy degenerált bázisnál vagyunk (hiszen vagy

$t_1=\beta_i/\alpha_i=0$, vagy $t_1=(\beta_i-u_i)/\alpha_i=0$ esetről van) szó és az x_j változó növelésével $w(t)$ romlani kezd.

A bemutatott alakok bővülhetnek azzal, hogy mindhárom esetben a maximumnál még vízszintes szakaszok is lehetnek.

Eredeti célunk az volt, hogy a kiválasztott x_j változót olyan szinten hozzuk be a bázisba, hogy ahhoz minél nagyobb $w(t)$ érték tartozzék. (A cél kitűzésénél még nem volt ismeretes $w(t)$ alakja, így nem lehetett tudni, hogy $w(t)$ maximumához báziscsere tartozik-e vagy sem.) Miután az eddigiek során végig a szimplex módszernek az egyedi felsőkorlát technikával kiegészített változatának megfelelő eseteket tárgyaltuk, meg kell említeni, hogy $w(t)$ maximalizálása során előfordulhat egy kedvezőnek tekinthető speciális eset. Ha ugyanis a belépő változó 1-es típusú, akkor előfordulhat, hogy a $w(t)$ maximumhelyét jelentő t_k érték eléri, vagy meghaladja a bázisba beléptetni kívánt x_j változó u_j egyedi felső korlátját: $t_k \geqq u_j$. A bevezetőben említett (i) feltétel miatt ekkor nem tudunk elmenni $w(t)$ maximumáig, hanem megállunk a $t=u_j$ helyen, viszont egy báziscsere nélküli iterációt hajtunk végre annak megfelelően, hogy az x_j változó átkerül alsó korlátjáról a felsőre. Ilyenkor új éta vektor [4] sem keletkezik, ami sok szempontból rendkívül előnyös jelenség [2].

Ha feltesszük, hogy nem az itt vázolt speciális esettel állunk szemben, akkor igaz a következő tétel:

5.2. TÉTEL. $w(t)$ a globális maximumát egy t_k töréspontban veszi fel és ez olyan báziscserét határoz meg, melynek során a bázist elhagyó i_k-adik bázisváltozó fizibilis értéken lép ki és a helyére belépő x_j változó is fizibilis értéket vesz fel.

A tétel első része $w(t)$ konkáv törtvonal függvény jellegéből (5.1. tétel) következik, a továbbiak pedig abból, hogy minden t_k ponthoz báziscsere tartozik, illetve ezekben a pontokban a pontokhoz tartozó — potenciális kilépő — változók fizibilis értéket vesznek fel.

Könnyen látható, hogy $w(t)$ maximumának meghatározását a t_k pontok és a hozzájuk tartozó α_{i_k} értékek ismeretében egyszerűen el lehet végezni.

Valójában azt nézzük meg, hogy $w(t)$ meredeksége hol vált előjelet, vagyis kiindulva az $r_1=-d_j$, $k=1$ állapotból addig folytatjuk az

$$r_{k+1} = r_k - |\alpha_{i_k}| \tag{5.2}$$

rekurziót, amíg $r_{k+1} \leqq 0$ nem teljesül. Az így adódó t_k pontban $w(t)$ felveszi maximumát. A továbbiakban a k indexszel a maximumot adó töréspontot jelöljük. Természetesen több pont is lehet, ahol ez igaz, pl. ha t_k-ban néhány pont egybeesik, vagy ha $w(t)$ vízszintes szakaszt tartalmaz a maximumnál. Ilyenkor a pivot sor kiválasztásában további szempontok érvényesítésére van lehetőség: 0 típusú változó bázisból való kilépésének preferálása, kedvezőbb nagyságú pivot elem kiválasztása stb.

A későbbiekben szükségünk lesz nemcsak egy maximumot adó t_k pont ismeretére, hanem ebben a pontban a $w(t_k)$ értékre is. Ezt természetesen kiszámíthatnánk $w(t)$ (4.2)-beli definíciója alapján, de ez viszonylag nehézkes lenne. Helyette az (5.2) rekurzió számításával egy időben és arra támaszkodva a $t_0=0$ jelölést bevezetve a

$$w(t_i) = w(t_{i-1}) + (t_i - t_{i-1}) r_i, \quad i = 1, \ldots, k \tag{5.3}$$

rekurzióval dolgozunk és így a k-adik lépésben éppen $w(t_k)$-t kapjuk. Ennek a belátásához elég $w(t)$ alakjára (5.1. tétel), a t_i és r_i értékek definíciójára gondolnunk.

Ennek a fejezetnek legfontosabb eredményeit ezek után az alábbiakban összegezhetjük:

— $w(t)$ szakaszonként lineáris konkáv függvény,
— $w(t)$ maximum helyét az (5.2) rekurzióval,
— $w(t)$ maximumát az (5.3) rekurzióval kapjuk meg.

Amennyiben a belépő változó negatív irányba mozdulva javít az infizibilitási helyzeten, akkor is igazak a fent elmondottak, csak értelemszerűen α_i helyett $-\alpha_i$-vel kell számolni és a transzformációs lépésnél előjelhelyesen kell eljárni.

Felmerülhet a kérdés, hogy mindennek mi az „ára", mennyivel kell több számítási munkát végezni, illetve mennyi új adatot kell tárolni. A T_a és T_f értékek számításánál szinte csak akkor akad — minimális — többletmunka, ha mindkét értéket meg kell határozni. Az így kapott értékeket nagyság szerint növekvő sorrendbe kell rendezni (4.5)-nek megfelelően. Ez összesen K szám rendezését jelenti, ami ugyan nem jár aritmetikai műveletek elvégzésével, de számítógépes futási időben némi pluszt jelenthet. Végül az (5.2) és az (5.3) rekurzió számolása még akkor sem számottevően hosszadalmas, ha a t_k sorozaton viszonylag messzire kell elmenni. Ennek a sorozatnak a lehetséges legnagyobb mérete $K=2(m-1)$, ami akkor fordulhat elő, ha — a célfüggvény változójától eltekintve — minden bázisváltozó 0 vagy 1 típusú, és mindegyikre ki kellett számítani T_a-t és T_f-et is. A számítási többletmunka végeredményben tehát nem lassítja észrevehetően az iterációs időt (hiszen minden műveletet a memóriában levő mennyiségekkel végzünk el), viszont az iterációk számának várható csökkenése miatt kevesebb mágneslemezes adatátvitelre lesz szükség, ami a megoldáshoz szükséges össz időigény szempontjából igen kedvező lehet. Néhány erre vonatkozó tapasztalatról a 8. fejezetben számolunk be.

A módszer tárolási többletigénye könnyen meghatározható. Miután $\max K= =2(m-1)$, ezért a T értékek tárolására egy ekkora területet kell fenntartani, továbbá szükséges egy ugyanekkora méretű permutáció-vektor is, ahol az átrendezés utáni állapotnak megfelelő indexeket helyezzük el. Ennek a két tömbnek a tárolása azonban nem jelent feltétlenül többlet memóriaigényt, mert a módosított szimplex módszer alkalmazása esetén a memória munkaterületén van olyan rész, amely a pivot lépés során egyébként kihasználatlan. Ez a megállapítás igaz az új módszernek „otthont" adó LIPROS [3] lineáris programozási programcsomagra is.

6. $w(t)$ néhány további tulajdonsága

Kapcsolat a hagyományos módszerrel

Először azt mutatjuk meg, hogy a báziscserének $w(t)$ maximalizálása útján történő meghatározása valójában annak a hagyományos pivotválasztási szabálynak az általánosítása, amely az első „ütköző" bázisváltozóig engedi növelni a belépő változót.

A hagyományos szabálynál a (4.3) és (4.4) képletekkel definiált nem-negatív T_a, illetve T_f értékek minimumát kell venni és ez lesz a belépő változó nagysága, ugyanakkor a minimumot szolgáltató bázisváltozó kilép a bázisból. Ez viszont

éppen annak felel meg, ha a $w(t)$ függvény maximalizálása helyett mindig csak a t_1 pontig mennénk, t_1 (4.5)-beli definíciójának megfelelően. Ily módon ha az 5.2. tétel szerint, $w(t)$ maximalizálásával határozzuk meg a báziscserét, akkor minden lépésben legalább annyit haladunk előre, mint a hagyományos módszerrel. Az új módszer konvergenciája ily módon következik a hagyományos módszer konvergenciájából, ha degeneráció nem lép fel.

A degeneráció vizsgálata

Tekintettel arra, hogy most felsőkorlát technikával van dolgunk, ezért a bázist akkor mondjuk degeneráltnak, ha valamelyik változója a fizibilis tartományának valamelyik határán (alsó vagy felső korlátján) van. Ekkor ugyanis a belépő változó értékének infinitezimális növelése esetén az ilyen bázisváltozók (felső korláton levő változónál $\alpha_i<0$ esetén, alsó korláton levőnél $\alpha_i>0$ esetén, tehát 0 típusúnál $\alpha_i\neq 0$-ra mindig) elkezdenek hozzájárulni az infizibilitások $w(t)$ mértékéhez. Miután a hagyományos pivotválasztási szabály nem engedi meg, hogy egyetlen bázisváltozó is átkerüljön az F halmazból az M vagy P halmazok valamelyikébe, ezért ilyenkor egy 0 szintű báziscserére kerülhet csak sor, aminek eredményeképpen az infizibilitások mértéke változatlan marad, tehát nem történik előrehaladás.

Az 5. fejezetben leírt módszer esetén a helyzet azonban másként alakulhat. Jelöljük l-lel a kiszámított 0 értékű T_a, illetve T_f értékek számát. Ez azt jelenti, hogy a (4.5) szerinti átrendezés után a

$$0 = t_1 = \ldots = t_l < t_{l+1} \leqq \ldots \leqq t_K \tag{6.1}$$

relációt kapjuk. Az (5.2) rekurzió értelmében $w(t)$ maximumát abban a t_k pontban veszi fel, melyre az teljesül, hogy

$$r_{k+1} \leqq 0 \quad \text{és} \quad r_k > 0,$$

illetve részletesen kiírva:

$$r_{k+1} = r_1 - \sum_{p=1}^{k} |\alpha_{i_p}| \leqq 0$$

és

$$r_k = r_1 - \sum_{p=1}^{k-1} |\alpha_{i_p}| > 0.$$

Tekintettel arra, hogy $r_1 = -d_j$, vagyis w(t)-nek a belépő x_j változó szerinti deriváltja, ezért a maximumot szolgáltató k indexet a

$$\begin{aligned} \sum_{p=1}^{k} |\alpha_{i_p}| &\geqq -d_j, \\ \sum_{p=1}^{k-1} |\alpha_{i_p}| &< -d_j \end{aligned} \tag{6.2}$$

összefüggések határozzák meg. Ha az így adódó k indexre a $k\leqq l$ reláció teljesül, akkor az új módszer szerint is csak 0 előrehaladás történik (3. ábra), ha azonban $k>l$, akkor $t_k>0$, így a degeneráció ellenére pozitív előrehaladás érhető el. Fentieket összefoglalva kimondható a következő tétel.

6.2. TÉTEL: Ha a 0 értéket adó T_a és T_f hányadosok (melyeket (6.1) szerint rendeztünk sorba) melletti α_i együtthatók abszolút értékeinek összege kisebb, mint $|d_j|$, akkor az 5.2. tétel által meghatározott báziscsere esetén — a degeneráció ellenére — az infizibilitások $w(t)$ mértéke

$$D = w(t_k) - w(0) > 0 \tag{6.3}$$

értékkel javul.

Egy ilyen báziscserének az eredményeképpen az infizibilitások száma esetleg növekedhet, a mértéke azonban biztosan javul. A hagyományos módszer melletti egyik legfőbb érv általában az, hogy veszélyes dolog az infizibilitások darabszámát növekedni hagyni, mert utána ezeket általában egyesével lehet csak újból megszüntetni. A mi esetünkben ez az érv azért ejthető el, mert igaz ugyan, hogy a darabszám egy-egy lépésben növekedhet, de egy másikban akár sokkal csökkenhet is, a mérték azonban végig monoton javul.

Numerikus stabilitás befolyásolása

Ismeretes, hogy a szimplex módszer numerikus problémáinak jó része a pivot elem numerikus szempontból esetleg nem ideális választásából fakad. A hagyományos pivot választási szabály esetén a pivot elem nagyságrendjének a figyelembevétele elég bonyolult feladat. Ha ugyanis a minimumot adó hányados egyértelműen adódik, akkor a nevező (pivot elem) esetén legfeljebb csak azt tudjuk megnézni, hogy a megengedett tűrési értéknél nem rosszabb-e, illetve ha igen, akkor mennyivel. Első esetben feltétlen elfogadjuk, második esetben esetleg kisebb engedményt teszünk a tűrési érték rovására, ha ez sem segít, akkor ezt az oszlopot ideiglenesen kizárjuk a jelöltek közül. Amikor a minimum egyszerre több soron adódik (pl. degenerációs nullák), akkor természetesen több a lehetőség a jó választásra. Számítógépes implementáció esetén ennél lényegesen bonyolultabb módon igyekeznek védekezni a rossz (túl kicsi, esetleg túl nagy) pivot elem ellen.

Az általunk bemutatott eljárás bizonyos esetekben egy igen egyszerű lehetőséget kínál megfelelő méretű pivot elem megtalálására. Arról van szó, hogy amennyiben $w(t)$ maximalizálása során $k \geqq 2$ értéket kaptunk és az ehhez tartozó pivot elem nem megfelelő, akkor k értékét csökkentve (ha kell, egészen $k=1$-ig) válogatni lehet a pivot elemek között. Ennek ára természetesen az, hogy ilyenkor $w(t)$-t általában már nem maximalizáljuk, hanem értelemszerűen csak szerényebb mértékben növeljük. Bizonyos esetekben azonban ezt érdemes vállalni, mert egy esetleges numerikus zavar az optimalizálás egész menetét komoly mértékben visszavetheti és ehhez képest egy nem optimális töréspont választása sokkal kisebb jelentőségű eseménynek számít.

7. Többszörös kiválasztás (multiple pricing)

Az eddigiekben azt az esetet vizsgáltuk, amikor valahogy eldöntöttük már, hogy melyik oszlopot léptetjük be a bázisba és ehhez határoztuk meg a bázist elhagyó oszlopot. Most egy kicsit visszalépünk a szimplex iteráció megtervezésének a vizsgálatában és azt, a gyakorlatban követett és jól bevált esetet tekintjük, amikor a mátrix egyszeri átnézése során több javító vektor jelöltet választunk ki *(multiple*

pricing) és azok közül további szempontok alapján határozzuk meg a bázisba ténylegesen beléptetendő oszlopot. Ezek a további szempontok bizonyos hierarchiát alkotnak, amelyek — részben a kidolgozó egyéni elgondolásait tükrözve — különbözőek lehetnek, azonban csaknem mindegyiknek a tetején a legnagyobb előrehaladás elve áll. Ennek a megokolására itt nem térünk ki, pusztán azt vizsgáljuk meg, hogy módszerünk hogyan illeszthető be az ilyen többszörös kiválasztási környezetbe.

Tegyük fel, hogy a kiválasztott oszlopok száma N. A hagyományos módszer először meghatározza minden oszlopra a pivot elemet és a hozzá tartozó θ_j ($j=1, \ldots, N$) hányadost, ami egyébként a belépő változó elmozdulásának nagyságát jelenti és amelyik oszlopra a $d_j\theta_j$ szorzat, vagyis a tényleges előreheladás a legnagyobb, azt vonja be a bázisba.

Az 5. fejezetben ismertetett módszer esetén a tényleges előrehaladást nem egy $d_j\theta_j$ jellegű lineáris kapcsolat határozza meg, hanem a j-edik kiválasztott oszlophoz tartozó $w_j(t)$ nem-lineáris függvény maximumának és kezdeti értékének a különbsége, D_j, amint az (6.3)-ból értelemszerűen kiolvasható. A kezdeti $w_j(0)$ érték minden j-re azonos (hiszen ez éppen az infizibilitások mértéke a soron következő iteráció előtt), $w_j(t)$ maximum értékét pedig egyszerűen megkapjuk az (5.3) rekurzióval. Így a j-edik oszlop bevonása esetén várható előrehaladás szinte számításaink melléktermékeként adódik, s a jelölt oszlopok közül könnyen kiválaszthatjuk a legkedvezőbbet.

A legnagyobb előrehaladás elvével kombinált többszörös kiválasztás esetén a bemutatott új módszer különösen az erősen degenerált esetekben tud igen hatásos lenni. A 6.2. tétel megmutatta, hogy egyetlen oszlop esetén milyen esély van a degenerált báziscsere elkerülésére. Ez a lehetőség most megsokszorozódik és ha a jelölt oszlopok között csak egy is $D_j>0$ értéket ad, akkor biztos, hogy nem degenerált báziscsere történik és ezzel együtt várhatóan a bázis degeneráltságának a foka is csökken elősegítve a későbbi hatékony iterációs lépéseket.

8. Néhány megjegyzés és kísérleti eredmény

1) A bemutatott eljárás hatékonyságát fokozni lehet azzal, hogy — most már ismerve működését — már az oszlopkiválasztásnál figyelemmel vagyunk $w(t)$, illetve a degeneráció szempontjaira. Az erre vonatkozó elgondolások ismertetése azonban nem szerepel jelen cikk célkitűzései között.

2) Szükségesnek látszik kitérni röviden egy — esetenként nem lényegtelen — mozzanatra. Fölmerülhet ugyanis az a gyanú, hogy egy fizibilis megoldás elérésének a „hajszolása" közben nem távolodunk-e el túlságosan az LP feladat optimumától, minthogy az igazi célfüggvényt teljesen figyelmen kívül hagyjuk. (Ez a kérdés természetesen nem merül fel, ha csak a feltételrendszer konzisztenciáját kell eldönteni.) Válaszként két dolgot kell megemlíteni. Először is semmi sem szól amellett, hogy most jobban távolodnánk el az optimumtól, mint más olyan első fázisú módszer esetén, amely szintén nem veszi figyelembe az igazi célfüggvényt. (Mellesleg a többszörös kiválasztáson belül a közel egyformán jó javító vektorok között könnyen preferencia adható annak, amelyik az igazi célfüggvény szerint kedvezőbbnek mutatkozik.) Másodszor: most is lehetőség van ún. összetett *(composite)* célfüggvény szerint végezni az oszlopkiválasztást, amelynek lényege az, hogy amíg sok javító

vektor kínálkozik — ez főleg kezdetben lehet igaz, — addig bizonyos súllyal az igazi célfüggvényt hozzáadjuk az infizibilitások mértékéhez és ezt az összetett célfüggvényt igyekszünk maximalizálni. Ennek részletes tárgyalásával itt nem foglalkozunk.

3) Az 5. fejezetben szereplő eljárás ismertetésénél szó volt arról, hogy a szükséges többletmunkát tulajdonképpen a kiszámított töréspontok helyeinek nagyság szerint növekvő sorrendbe történő rendezése jelenti. Amennyiben K (az összes töréspontok száma) nagy, úgy ez a többletmunka már észrevehető időt vehet igénybe. Egy egyszerű tapasztalat felhasználásával azonban elkerülhető a „fölösleges" rendezés. Arról van szó, hogy a szimplex eljárás kezdeti szakaszában K értéke általában kicsinek adódik (tipikus tartomány R10-en közepes méretű feladatokra: $5 \leqq K \leqq 20$), és ilyenkor a maximumot szolgáltató k index $K/2$ körül ingadozik. Az iterációk előrehaladtával K értéke hajlamos a növekedésre (tipikus tartomány: $20 \leqq K \leqq 80$), ugyanakkor k értéke 5 (gyakran 3) alá esik. Ebben a helyzetben természetesen fölösleges a K érték teljes rendezése, hiszen csak a néhány legkisebbre van szükségünk. E helyett az a helyes, ha olyan rendező eljárást használunk, amelyik alulról fokozatosan építi fel a rendezettséget és az i-edik lépés után az első i érték helyes sorrendbe kerül. Így a rendezés lépéseivel párhuzamosan tudjuk számolni az (5.2) és (5.3) rekurziót és abban a pillanatban leállunk a rendezéssel, mihelyt az (5.2) rekurzió eredményének ellenőrzése jelzi egy optimális megoldás (töréspont) megtalálását. A K és k-ra vonatkozó tapasztalatok tükrében ilyen módon lényeges számítási megtakarítás érhető el olyan rendezési eljárásokkal szemben, melyek csak a legvégén szolgáltatnak egy helyesen rendezett számsort.

4) A bemutatott első fázisú eljárás implementációja elkészült és beépült a LIPROS lineáris programozási programcsomagba. Ez a programcsomag elsősorban az R sorozat legkisebb gépeire, az R10 és R12-re készült. Ezeknek a gépeknek a központi egysége gyors, így az aritmetikai többletmunka általában kifizetődő, ha ezzel meg lehet takarítani valamit a háttértárolókkal történő adatátvitelből. Ez eredményezte azt, hogy a $w(t)$-vel kapcsolatos számítások alig észrevehetően növelték csak az iterációs időket.

Az új eljárással kapcsolatos összehasonlító vizsgálatokat úgy végeztük el, hogy a LIPROS PIVOT1 nevű régi pivot választó szubrutinja helyére betettük az új első fázisú eljárást megvalósító PI1 nevű szubrutint. Ez a PI1-re nézve azt a hátrányt jelentette, hogy kénytelen volt azokkal a kiválasztott oszlopokkal dolgozni, amelyeket a PIVOT1 számára kidolgozott eljárás határozott meg.

Az alábbi táblázatban néhány jellegzetesnek talált feladatra vonatkozó futási statisztikai adatot mutatunk be. A futások PIVOT1 és PI1 esetében teljesen azonos körülmények (futási paraméterek stb.) között zajlottak le. Mindegyik esetben a tisztán logikai változókból álló bázisból indultunk ki. A táblázat adatai az első fázisban végzett iterációk számát jelentik, a méretnél először a sorok száma (m), majd az oszlopok száma (n) szerepel.

Az 1. és 5. feladatban a változók egy részének volt egyedi felső korlátja.

A feladatokra vonatkozó megjegyzéseken túl néhány, a megoldás menetére vonatkozó megjegyzést kell tenni.

A 3. feladat csupa $\geqq$ típusú feltételből állt. Az infizibilitások mértékének monoton javulása mellett az infizibilitások száma többnyire „drasztikus" módon

Feladat		PIVOT1	PI1	Megjegyzés
szám	méret			
1.	17×12	18	15	10 változó felső korlátos, az induló bázis degeneráltsági foka: 50%
2.	62×70	43	37	25 db egyenlőség a feltételek között
3.	61×10	42	12	Induló bázis minden változója infizibilis
4.	100×130	93	74	50 db egyenlőség a feltételek között
5.	170×120	—	278	Induló bázis degeneráltsági foka: 96%

csökkent (1 lépésben 10-20 infizibilitás is megszűnt), de két esetben kicsit nőtt. Összességében úgy tűnt, hogy az új módszer potenciális előnyei ennél a feladatnál az átlagosnál jobban érvényesültek.

Az 5. feladatot PIVOT1-gyel nem sikerült megoldani. Az eljárás az iterációk hosszú során nem tudott elmozdulni az erősen degenerált induló bázisról és — bár ciklusba nem esett — a numerikus hibák kedvezőtlen halmozódása miatt (az R10-en 4 *byte*-os lebegőpontos aritmetikát használunk) az adott keretek között a feladatot nem tudta megoldani, amint ezt az ötszázadik iteráció körül kijelezte. A PI1 eljárás is sokat „birkózott" a degenerált induló bázissal, de a 205. iterációban talált egy pozitív előrehaladást, mégpedig annak az árán, hogy az infizibilitások száma 1-ről felment 5-re. Ezután már a megoldás simán zajlott tovább egészen a 278. lépésben bekövetkezett befejezésig. Numerikus pontatlanság egyetlen alkalommal sem vetette vissza a számítások menetét, minden közbülső újrainvertálás jól reprodukálta az invertálás előtti állapotot és a végeredmény is pontosnak adódott.

IRODALOM

[1] HADLEY, G., *Linear Programming* (Addison—Wesley, 1962).
[2] MAROS, I., „Adaptív elemek a lineáris programozásban", *Alk. Mat. Lapok* 2 (1976).
[3] MAROS, I., „Lineáris programozási programcsomag R10, R12 számítógépekre", *Struktúra* 9 (1979).
[4] ORCHARD—HAYS, W., *Advanced Linear Programming Computing Techniques* (McGraw, 1968).
[5] PRÉKOPA, A., *Lineáris programozás I.*, Bolyai János Matematikai Társulat, 1968.

(Beérkezett: 1980. március 17.)

MAROS ISTVÁN
SZÁMÍTÓGÉPALKALMAZÁSI KUTATÓ INTÉZET
1536 BUDAPEST, PF. 227.

DETERMINING THE OUTGOING VARIABLE IN PHASE I OF THE SIMPLEX METHOD

I. MAROS

Phase I of the simplex method serves for finding a feasible solution to linear programming (LP) problems but it also can be used for the same purpose in the case of other problems having linear inequalities as constraints, or even for detecting the consistency of systems of linear equalities/inequalities.

In the case of LP problems the traditional phase I methods work under the following conditions:

(i) The incoming variable takes on a feasible value.

(ii) The feasible basic variables remain feasible after transformation (the number of infeasibilities does not increase).

(iii) The outgoing variable leaves the basis at a feasible level (lower or upper bound).

By relaxing condition (ii) new possibilities appear still within the frame of the simplex method. The paper presents a procedure based on this relaxation by which more efficient iterations can be expected in phase I — especially in the presence of degeneracy — simply owing to the new way of determining the outgoing variable. Some computational experiences are also reported.

Alkalmazott Matematikai Lapok 6 (1980) 17—26

A NEMLINEÁRIS GÖRBEILLESZTÉS EGY ÚJ MÓDSZERE

KUTAS TIBOR

Budapest

A dolgozatban a nemlineáris görbeillesztés numerikus megoldásának egy új módszerét ismertetem. A 3. fejezetben leírt algoritmus, hasonlóan a többihez, a *Gauss—Newton-módszer* hibáit igyekszik kiküszöbölni. Az új módszer elsősorban olyan esetekben alkalmazható, amikor a minimalizálandó függvény kiszámítása nehézkes — például egy differenciálegyenlet kiintegrálását igényli —, így egy lépésben csak egyszer akarjuk kiszámítani az értékét. A 4. fejezet különböző tesztfeladatokon hasonlítja össze azokat az algoritmusokat, amelyek egy iterációs lépésben egyszer, vagy legalábbis kevésszer számítják ki a függvény értékét.

1. Bevezetés

A paraméterbecslés egy a gyakorlatban sokszor előforduló esete a felületilletve görbeillesztés a legkisebb négyzetek módszerével. A probléma a következő: adott n számú k-dimenziós vektor $(\mathbf{x}_i)$ és ugyanennyi valós szám (y_i), valamint egy $f: R_k \to R_1$ függvény, melyekre az alábbi összefüggés áll fenn

$$y_i = f(\mathbf{x}_i, \vartheta) + \varepsilon_i,$$

ahol ϑ l-dimenziós vektor, ε_i 0 várható értékű, σ szórású, normális eloszlású valószínűségi változó, és feltételezzük, hogy függetlenek. A feladat a ϑ paramétervektor becslése. Ennek kézenfekvő módja a legkisebb négyzetek módszere, ami azt jelenti, hogy olyan paramétervektort keresünk, amelyre az eltérések négyzetösszege minimális, azaz minimalizáljuk a $H(\vartheta)$ függvényt, ahol

$$H(\vartheta) = \sum_{i=1}^{n} (f(\mathbf{x}_i, \vartheta) - y_i)^2.$$

Abban az esetben, ha az f függvény a ϑ vektorváltozóban lineáris, akkor az ε_i valószínűségi változókra kirótt feltételek miatt az optimális ϑ vektor torzítatlan becslés, és a *Gauss—Markov-tétel* állítása szerint minimális szórású is (lásd pl. [1], [9]). Nemlineáris esetben egyik tulajdonságot sem tudjuk biztosítani, ennek ellenére ekkor is gyakran használják ezt a módszert paraméterbecslésre. A sztochasztikus feltételeket figyelmen kívül hagyva sokszor nem a ϑ paramétervektor becslése a fő feladat, hanem például interpoláció vagy esetleg extrapoláció.

Ebben a dolgozatban nem vizsgáljuk a paraméterbecslés statisztikai hátterét, itt csak a probléma numerikus megoldásával foglalkozunk. A nemlineáris görbeillesztést az emeli ki az általános nemlineáris programozási feladatok közül, hogy a változószám alacsony (legtöbbször tíz alatti), másrészt a $H(\vartheta)$ függvény alakját

az algoritmusok konstruálásánál jól ki lehet használni, ugyanekkor azonban nem konvex, sőt jól ellenőrizhető feltételt sem lehet adni (a lineáris eset kivételével), amely biztosítaná konvexségét.

A dolgozat 2. fejezete a nemlineáris görbeillesztés általános módszereiről szól, a 3. fejezet pedig egy olyan új algoritmust ismertet, amelynek célja, hogy olyan speciális esetben, amikor a függvény kiszámítása bonyolult (például egy differenciálegyenlet megoldását igényli), minél kevesebb függvényérték számolással meg lehessen oldani a feladatot. A 4. fejezet a numerikus tapasztalatokat foglalja össze.

Néhány jelölés: az y_i valós számokat tekintsük egyetlen n-dimenziós vektornak és hasonlóan az $f(\mathbf{x}_i, \vartheta)$ értékeket is. Jelölje ezeket $\mathbf{Y}$ és $\mathbf{f}$, azaz

$$\mathbf{f}(\vartheta) = [f(\mathbf{x}_1, \vartheta), \ldots, f(\mathbf{x}_n, \vartheta)]^T,$$

$$\mathbf{Y} = [y_1, \ldots, y_n]^T,$$

továbbá az így definiált $\mathbf{f}(\vartheta)$ leképezés *Jacobi-mátrixát* jelölje $\nabla\mathbf{f}(\vartheta)$, azaz

$$[\nabla\mathbf{f}(\vartheta)]_{ij} = \frac{\partial f(\mathbf{x}_i, \vartheta)}{\partial \vartheta_j}.$$

2. A görbeillesztés általános módszerei

Az alább leírt algoritmusok mind a *Gauss—Newton-módszer* különböző módosításai, melyek az eredeti algoritmus hibáit igyekeznek kiküszöbölni.

Számítsuk ki a $H(\vartheta)$ függvény gradiensét és *Hesse-mátrixát:*

$$\nabla H(\vartheta) = 2\nabla^T\mathbf{f}(\vartheta)(\mathbf{f}(\vartheta)-\mathbf{Y})$$

$$\nabla^2 H(\vartheta) = 2\sum_{i=1}^{n}\left[(f(\mathbf{x}_i, \vartheta)-y_i)\nabla^2_\vartheta f(\mathbf{x}_i, \vartheta)\right]+2\nabla^T\mathbf{f}(\vartheta)\,\nabla\mathbf{f}(\vartheta).$$

Ismeretes, hogy egy általános $g(\mathbf{x})$ függvény feltétel nélküli minimalizálására a leggyorsabb konvergenciájú a *Newton-módszer* ([1])

$$\mathbf{x}_{r+1} = \mathbf{x}_r-(\nabla^2 g(\mathbf{x}_r))^{-1}\nabla g(\mathbf{x}_r).$$

A módszer hátránya, hogy analitikusan ki kell számolni a másodrendű parciális deriváltakat. A görbeillesztés esetében a $H(\vartheta)$ függvény *Hesse mátrixát* közelíti a $2\nabla^T\mathbf{f}(\vartheta)\cdot\nabla\mathbf{f}(\vartheta)$ mátrix. Ha az $f(\mathbf{x}, \vartheta)$ függvény a ϑ változóban lineáris, akkor nemcsak közelíti, de azonos vele. A közelítés annál jobb, minél közelebb van a mért y_i értékhez a számított $f(\mathbf{x}_i, \vartheta)$ érték; amennyiben az optimum érték nulla, az iterációk során a $H(\vartheta)$ függvény *Hesse-mátrixát* is közelítjük. Ha a *Newton-módszerben* a *Hesse mátrix* helyébe a fenti közelítést írjuk be, kapjuk a nemlineáris görbeillesztés legrégibb algoritmusát, a *Gauss—Newton-módszert* ([1], [2]):

$$\vartheta_{r+1} = \vartheta_r-(\nabla^T\mathbf{f}(\vartheta_r)\nabla\mathbf{f}(\vartheta_r))^{-1}\nabla^T\mathbf{f}(\vartheta_r)(\mathbf{f}(\vartheta_r)-\mathbf{Y}).$$

Ismeretes, hogy a gradiens típusú minimalizáló algoritmusoknál egy irány akkor jó, ha 90°-nál kisebb szöget zár be a függvény negatív gradiensével, azaz a módosító mátrix pozitív szemidefinit ([1]). Ez ebben az esetben teljesül, hiszen a $(\nabla^T\mathbf{f}\nabla\mathbf{f})$

mátrix mindig pozitív szemidefinit, és ha invertálható, az inverz pozitív definit lesz. Az algoritmust levezethetjük úgy is, hogy nem a $H(\vartheta)$ függvényt fejtjük sorba a ϑ_r pont környezetében, hanem az $f(\mathbf{x}_i, \vartheta)$ függvényeket fejtjük sorba lineáris tagig, így közelítve kvadratikus függvénnyel a $H(\vartheta)$ függvényt. A módszernek két hátránya van: az invertálandó mátrix nem invertálható vagy közel szinguláris (a legkisebb sajátérték közel van a nullához); az algoritmus nem monoton, azaz nem áll fenn minden lépésben a következő egyenlőtlenség

$$H(\vartheta_{r+1}) < H(\vartheta_r).$$

Ez utóbbi hátrányt küszöböli ki *Hartley módszere,* amely abban különbözik a *Gauss—Newton-algoritmustól,* hogy minden lépésben irány menti minimalizálást végez ([4]).

A *Gauss—Newton-módszer* mindkét hátrányán próbál segíteni LEVENBERG és MARQUARDT ötlete ([5], [6]) (az irodalomban többnyire mint *Marquardt-módszer* ismeretes). Az eredeti $H(\vartheta)$ függvény helyett minimalizáljuk a következő $M(\vartheta, \lambda)$ függvényt

$$M(\vartheta, \lambda) = H(\vartheta) + \lambda \|\vartheta\|^2 = \sum_{i=1}^{n} (f(\mathbf{x}_i, \vartheta) - y_i)^2 + \lambda \cdot \sum_{j=1}^{l} \vartheta_j^2,$$

ahol λ nemnegatív valós szám. Természetesen az iteráció során a λ szám nullához tart, így biztosítva, hogy az eredeti feladat optimumát kapjuk. Ha erre az $M(\vartheta, \lambda)$ függvényre alkalmazzuk ugyanazt a gondolatmenetet, mint a *Gauss—Newton-módszernél,* a következő iterációt kapjuk

$$\vartheta_{r+1} = \vartheta_r - [\nabla^T \mathbf{f}(\vartheta_r) \nabla \mathbf{f}(\vartheta_r) + \lambda \mathbf{I}]^{-1} \nabla^T \mathbf{f}(\vartheta_r)(\mathbf{f}(\vartheta_r) - \mathbf{Y}).$$

Mivel a $(\nabla^T \mathbf{f} \nabla \mathbf{f})$ mátrix pozitív szemidefinit, ezért pozitív λ szám esetén a $(\nabla^T \mathbf{f} \nabla \mathbf{f} + \lambda \mathbf{I})$ mátrix pozitív definit, így invertálható és az inverz is pozitív definit, tehát az irány ez előző megjegyzés értelmében megfelelő. A *Marquardt-módszert* olyan módon is be lehet vezetni, hogy a *Gauss—Newton-módszernél* az invertálandó mátrix főátlójához hozzáadunk egy pozitív számot, ezzel stabilissá téve a mátrixinverziót. Ezt a módszert *Tyihonov-regularizáció* néven alkalmazzák lineáris egyenletrendszerek megoldásánál.

Vizsgáljuk meg, hogy a λ paraméter függvényében hogyan változik az új iterációs pont! A kérdést egy kicsit általánosabban nézzük meg. Tekintsük az

$$\mathbf{A}\mathbf{z} = \mathbf{b} \tag{2.1}$$

lineáris egyenletrendszert, ahol $\mathbf{A}$ szimmetrikus pozitív szemidefinit $l \times l$ dimenziós mátrix, $\mathbf{b}$ és $\mathbf{z}$ l-dimenziós vektorok. A (2.1) egyenletrendszer helyett oldjuk meg a

$$(\mathbf{A} + \lambda \mathbf{I})\mathbf{z} = \mathbf{b}$$

egyenletrendszert, ahol λ pozitív valós szám. Definiáljuk a $\mathbf{z}(\lambda)$ l-dimenziós görbét a következőképpen

$$\mathbf{z}(\lambda) = (\mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{b}. \tag{2.2}$$

Mint fent láttuk, pozitív λ szám esetén az inverz létezik, így a fenti definíció értelmes. Az inverz folytonossága és differenciálhatósága következtében a görbe folytonos és differenciálható. Ha az $\mathbf{A}$ mátrix invertálható, akkor $\mathbf{z}(0)$ vektor is létezik és az eredeti feladat megoldása.

2.1. TÉTEL. Ha a λ szám tart végtelenhez, akkor a (2.2)-ben definiált görbe a nulla vektorhoz tart, azaz

$$\lim_{\lambda\to\infty} \|\mathbf{z}(\lambda)\| = 0.$$

Bizonyítás. A tételt indirekt módon látjuk be, tegyük fel, hogy az állítás nem igaz. Ekkor létezik olyan λ_i végtelenhez tartó sorozat, $i=1, \ldots, n, \ldots$, hogy fennáll a következő egyenlőtlenség:

$$\|\mathbf{z}(\lambda_i)\| \geqq \varepsilon > 0.$$

Felírhatjuk az alábbi egyenlőtlenségsorozatot

$$\|\mathbf{b}\| = \|(\mathbf{A}+\lambda_i\mathbf{I})\mathbf{z}(\lambda_i)\| \geqq \lambda_i\|\mathbf{z}(\lambda_i)\| - \|\mathbf{A}\mathbf{z}(\lambda_i)\|,$$

$$\lambda_i\|\mathbf{z}(\lambda_i)\| - \|\mathbf{A}\mathbf{z}(\lambda_i)\| \geqq (\lambda_i - \|\mathbf{A}\|)\|\mathbf{z}(\lambda_i)\|.$$

Mivel λ_i végtelenhez tart, ezért $(\lambda_i - \|\mathbf{A}\|)$ is tart végtelenhez, s mivel $\|\mathbf{z}(\lambda_i)\|$ érték egy pozitív ε szám fölött van, így a szorzat is végtelenhez tart. Összehasonlítva az egyenlőtlenségsorozat bal és jobb oldalát, a

$$\|\mathbf{b}\| \geqq (\lambda_i - \|\mathbf{A}\|)\|\mathbf{z}(\lambda_i)\|$$

ellentmondáshoz jutunk. Ezzel állításunkat beláttuk.

2.2. TÉTEL.

$$\lim_{\lambda\to\infty} \frac{\mathbf{z}^T(\lambda)\mathbf{b}}{\|\mathbf{z}(\lambda)\| \cdot \|\mathbf{b}\|} = 1,$$

azaz a $\mathbf{z}(\lambda)$ megoldásvektornak a $\mathbf{b}$ vektorral bezárt szöge nullához tart.

Bizonyítás. A $\mathbf{z}(\lambda)$ vektor (2.2) definícióját felhasználva a következő egyenlőségeket írhatjuk fel:

$$\cos\alpha(\lambda) = \frac{\mathbf{z}^T(\lambda)\mathbf{b}}{\|\mathbf{z}(\lambda)\| \cdot \|\mathbf{b}\|} = \frac{\mathbf{z}^T(\lambda)(\mathbf{A}+\lambda\mathbf{I})\mathbf{z}(\lambda)}{\|\mathbf{z}(\lambda)\| \cdot \|\mathbf{b}\|} = \frac{\mathbf{z}^T(\lambda)\mathbf{A}\mathbf{z}(\lambda)}{\|\mathbf{z}(\lambda)\| \cdot \|\mathbf{b}\|} + \lambda\frac{\|\mathbf{z}(\lambda)\|^2}{\|\mathbf{z}(\lambda)\| \cdot \|\mathbf{b}\|}.$$

Felhasználva a $\mathbf{z}^T(\lambda)\mathbf{A}\mathbf{z}(\lambda) \geqq -\|\mathbf{A}\|\,\|\mathbf{z}(\lambda)\|^2$ egyenlőtlenséget, és $\|\mathbf{z}(\lambda)\|$-val egyszerűsítve a következő egyenlőtlenséget kapjuk

$$\cos\alpha(\lambda) \geqq -\frac{\|\mathbf{A}\|}{\|\mathbf{b}\|} \cdot \|\mathbf{z}(\lambda)\| + \lambda \cdot \frac{\|\mathbf{z}(\lambda)\|}{\|\mathbf{b}\|}.$$

Nézzük külön a második tagot; a (2.2) definíciót felhasználva a következő egyenlőtlenséget írhatjuk fel

$$\lambda\frac{\|\mathbf{z}(\lambda)\|}{\|\mathbf{b}\|} = \lambda\frac{\|\mathbf{z}(\lambda)\|}{\|(\mathbf{A}+\lambda\mathbf{I})\mathbf{z}(\lambda)\|} \geqq \lambda\frac{\|\mathbf{z}(\lambda)\|}{\lambda\|\mathbf{z}(\lambda)\| + \|\mathbf{A}\mathbf{z}(\lambda)\|} \geqq \frac{\lambda \cdot \|\mathbf{z}(\lambda)\|}{\lambda\|\mathbf{z}(\lambda)\| + \|\mathbf{A}\| \cdot \|\mathbf{z}(\lambda)\|}.$$

Itt az egyszerűsítéseket elvégezve és visszaírva az eredeti egyenlőtlenségbe a következőt kapjuk:

$$\cos\alpha(\lambda) \geqq -\frac{\|\mathbf{A}\|}{\|\mathbf{b}\|} \cdot \|\mathbf{z}(\lambda)\| + \frac{\lambda}{\lambda+\|\mathbf{A}\|}.$$

A 2.1. tételben beláttuk, hogy $\|\mathbf{z}(\lambda)\| \to 0$, ha $\lambda \to \infty$, ezért az első tag nullához tart, míg a második nyilvánvalóan 1-hez. Ezzel állításunkat beláttuk.

Nézzük meg, mit jelent ez a két tétel a *Marquardt-módszer* esetében. Ha a λ paramétert növeljük, az új iterációs pont a régi iterációs ponthoz közeledik (2.1. tétel), és az irány a negatív gradiens irányához tart (2.2. tétel). Ez biztosítja, hogy ha elég nagy λ számot választunk, a $H(\mathfrak{P})$ függvény értéke csökkenni fog. Erre a tényre alapozódik a MARQUARDT által javasolt algoritmus. Legyen

$$\lambda_{r+1} = \begin{cases} \lambda_r/v, & \text{ha } H(\mathfrak{P}(\lambda_r/v)) < H(\mathfrak{P}_r), \\ \lambda_r, & \text{ha } H(\mathfrak{P}(\lambda_r/v)) \geqq H(\mathfrak{P}_r) > H(\mathfrak{P}(\lambda_r)), \\ \lambda_r \cdot v^i, & \text{ahol } i \text{ a legkisebb pozitív egész, melyre} \\ & H(\mathfrak{P}(\lambda_r \cdot v^{i-1})) \geqq H(\mathfrak{P}_r) > H(\mathfrak{P}(\lambda_r \cdot v^i)), \end{cases} \tag{2.3}$$

ahol

$$\mathfrak{P}(\lambda) = \mathfrak{P}_r - [\nabla_r \mathbf{f}(\mathfrak{P}_r) \nabla \mathbf{f}(\mathfrak{P}_r) + \lambda \mathbf{I}]^{-1} \nabla^T \mathbf{f}(\mathfrak{P}_r)(\mathbf{f}(\mathfrak{P}_r) - \mathbf{Y})$$

és v előre rögzített egynél nagyobb szám.

Az R. R. MEYER által javasolt algoritmus ([7]) kombinálja MARQUARDT és HARTLEY ötletét, nevezetesen a λ paramétert iterációs lépésenként csökkenti, és az így kapott irányban irány menti minimalizálást végez.

A numerikus tapasztalatok azt mutatják, hogy az esetek egy részében a *Gauss—Newton-módszer* nem konvergál, amikor más módszerek megtalálják az optimumot, de ha konvergál, akkor az összes módszer közül a leggyorsabban, a legkevesebb lépésben, a legkevesebb függvényérték-számolással találja meg az optimum pontot ([2], [3]).

3. A módosított Marquardt-módszer

A nemlineáris görbeillesztésnél előfordulhat olyan eset, amikor a $H(\mathfrak{P})$ függvény kiszámítása bonyolult, számítógépes megvalósítás esetén sok gépidőt vesz igénybe. Például abban az esetben, ha az $f(\mathbf{x}, \mathfrak{P})$ függvényt egy analitikusan nem megoldható differenciálegyenlet adja meg. Ilyen esetekben a $H(\mathfrak{P})$ függvény kiszámítása egy pontban nagyságrenddel több időt vehet el, mint a lineáris algebrai műveletek. Az is világos, hogy ilyen esetekben irány menti minimalizálást tartalmazó algoritmusok, mint esetünkben *Hartley és Meyer módszere*, szóba sem jöhetnek. Az algoritmusokat áttekintve a *Gauss—Newton-módszer* látszik a legalkalmasabbnak, hiszen gyors és egy lépésben csak egyszer kell kiszámolni a $H(\mathfrak{P})$ függvény értékét. Hátránya, hogy sok esetben nem konvergál. Ezért olyan módszert próbáltam konstruálni, amely lehetőleg közelíti a *Gauss—Newton-algoritmust*, lépésenként csak egyszer kell a $H(\mathfrak{P})$ függvényt kiértékelni, de az invertálandó mátrix a *Marquardt-módszerhez* hasonlóan mindig pozitív definit. Az algoritmus a következő:

$$\mathfrak{P}_{r+1} = \mathfrak{P}_r - [(\nabla^T \mathbf{f}(\mathfrak{P}_r) \nabla \mathbf{f}(\mathfrak{P}_r) + \lambda \mathbf{I})^{-1} + \tag{3.1}$$

$$+ \alpha(\nabla^T \mathbf{f}(\mathfrak{P}_r) \nabla \mathbf{f}(\mathfrak{P}_r) + \lambda \mathbf{I})^{-1} \cdot (\nabla^T \mathbf{f}(\mathfrak{P}_r) \nabla \mathbf{f}(\mathfrak{P}_r) + \lambda \mathbf{I})^{-1}] \cdot \nabla^T \mathbf{f}(\mathfrak{P}_r) \cdot (\mathbf{f}(\mathfrak{P}_r - \mathbf{Y})).$$

A *Marquardt-módszernél* az új iterációs pontot egy λ paraméterű görbén keressük meg a (2.3) algoritmus alapján. Az új módszer lényege, hogy csökkenő λ_i paramétrsorozat mellett nem a görbe λ_i paraméterű pontját választjuk, hanem a görbe ezen pontjából kiinduló érintőn közelítjük a *Gauss—Newton-módszer* iterációs

pontját. Az α paramétert úgy választjuk, hogy legalábbis lineáris esetben (az $\mathbf{f}(\vartheta)$ függvény lineáris) a $H(\vartheta)$ függvény értéke minimális legyen, s így közel lesz a *Gauss—Newton-módszer* adta iterációs ponthoz. Néhány egyszerűsítő jelölés: mivel lineáris esetben a $\mathbf{f}(\vartheta)$ függvény gradiense nem függ a ϑ vektor értékétől, ezért csak a $\nabla\mathbf{f}$ jelölésre rövidítjük; továbbá legyen

$$\mathbf{A} = \nabla^T\mathbf{f}\nabla\mathbf{f}, \tag{3.2}$$

$$\mathbf{B} = (\mathbf{A}+\lambda\mathbf{I})^{-1}+\alpha(\mathbf{A}+\lambda\mathbf{I})^{-1}\cdot(\mathbf{A}+\lambda\mathbf{I})^{-1}. \tag{3.3}$$

Ezekkel a jelölésekkel a (3.1)-ben leírt algoritmus a következőképp adható meg:

$$\vartheta_{r+1} = \vartheta_r - \mathbf{B}\nabla^T\mathbf{f}(\nabla\mathbf{f}\vartheta_r - \mathbf{Y}). \tag{3.4}$$

A $H(\vartheta)$ értéke a következőképp számolható, behelyettesítve a (3.4) képletet és átrendezve:

$$H(\vartheta_{r+1}) = \|\nabla\mathbf{f}\cdot\vartheta_r - \mathbf{Y}\|^2 - [\nabla^T\mathbf{f}(\nabla\mathbf{f}\vartheta_r - \mathbf{Y})]^T[2\mathbf{B}-\mathbf{BAB}][\nabla^T\mathbf{f}(\nabla\mathbf{f}\vartheta_r - \mathbf{Y})].$$

Itt az összeg első tagja a $H(\vartheta)$ függvény értéke az előző iterációs pontban. Ha az α paraméter értékét úgy választjuk meg, hogy a $(2\mathbf{B}-\mathbf{BAB})$ mátrix pozitív definit, akkor a módszer monoton csökkenő. De explicite ki is számolhatjuk, hogy milyen α paraméterértékre lesz a második tag (mint kivonandó) maximális. A

$$\frac{d}{d\alpha}\{[\nabla^T\mathbf{f}\cdot(\nabla\mathbf{f}\cdot\vartheta_r - \mathbf{Y})]^T[2\mathbf{B}-\mathbf{BAB}]\cdot[\nabla^T\mathbf{f}(\nabla\mathbf{f}\cdot\vartheta_r - \mathbf{Y})]\} = 0 \tag{3.5}$$

egyenletet kell megoldanunk. A szorzatban csak a $\mathbf{B}$ mátrix függ az α paraméter értékétől, így felhasználva a

$$\frac{d\mathbf{B}}{d\alpha} = (\mathbf{A}+\lambda\mathbf{I})^{-1}\cdot(\mathbf{A}+\lambda\mathbf{I})^{-1}$$

azonosságot, a (3.5) egyenlet bal oldalára a következő kifejezést kapjuk:

$$\mathbf{h}^T[2\cdot(\mathbf{A}+\lambda\mathbf{I})^{-1}\cdot(\mathbf{A}+\lambda\mathbf{I})^{-1} - (\mathbf{A}+\lambda\mathbf{I})^{-1}(\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{A}\cdot\mathbf{B} - \mathbf{B}\cdot\mathbf{A}\cdot(\mathbf{A}+\lambda\mathbf{I})^{-1}(\mathbf{A}+\lambda\mathbf{I})^{-1}]\cdot\mathbf{h},$$

ahol

$$\mathbf{h} = \nabla^T\mathbf{f}(\nabla\mathbf{f}\vartheta_r - \mathbf{Y}).$$

Kiemelve balról és jobbról az $(\mathbf{A}+\lambda\mathbf{I})^{-1}(\mathbf{A}+\lambda\mathbf{I})^{-1}$ mátrixot és bevezetve a $\mathbf{g}=(\mathbf{A}+\lambda\mathbf{I})^{-1}(\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{h}$ jelölést, az alábbi egyenletet kapjuk:

$$2\mathbf{g}^T[(\lambda-\alpha)\mathbf{A}+\lambda^2\mathbf{I}]\mathbf{g} = 0.$$

Ezt már könnyű megoldani, a megoldás:

$$\alpha = \lambda+\lambda^2\frac{\|\mathbf{g}^2\|}{\mathbf{g}^T\mathbf{A}\mathbf{g}}. \tag{3.6}$$

Az α paraméter szerinti második derivált értéke ebben a pontban a $-2\mathbf{g}\mathbf{A}\mathbf{g}$ szorzattal egyenlő. Az $\mathbf{A}$ mátrix pozitív szemidefinitsége miatt a szorzat értéke nempozitív, ezért a derivált zérushelye lokális, sőt mivel másodfokú függvényről van szó, globális

maximum. A (3.6) képletben az összeg második tagjának kiszámítása nehézkes, másrészt a λ paraméter értéke kicsi, így a λ értéke elég jól közelíti az α paraméter optimális értékét.

Hátra van annak bizonyítása, hogy az α paraméter ilyen választása mellett a $2\mathbf{B}-\mathbf{BAB}$ mátrix pozitív definit, s így az algoritmus monoton csökkenő. Ismeretes, hogy egy szimmetrikus mátrix felírható a következő alakban:

$$\mathbf{A} = \mathbf{U}^T\mathbf{D}\mathbf{U}, \tag{3.7}$$

ahol $\mathbf{U}$ unitér és $\mathbf{D}$ diagonál mátrix, azaz

$$[\mathbf{D}]_{i,j} = \begin{cases} d_i, & i = j, \\ 0, & i \neq j. \end{cases}$$

A (3.2)-ben definiált $\mathbf{A}$ mátrix szimmetrikus, és mivel pozitív szemidefinit, ezért a d_i értékei nemnegatívak, és ez egyszerűbb jelölés kedvéért feltehetjük, hogy monoton növekedők, azaz

$$d_i \leqq d_j, \quad \text{ha} \quad i \leqq j.$$

Ezzel a felírással könnyű hasonló alakra hozni a (3.3)-ban definiált $\mathbf{B}$ mátrixot, amely a (3.7) jelölésével a következő alakú:

$$\mathbf{B} = \mathbf{U}^T \begin{bmatrix} \dfrac{1}{d_1+\lambda} + \dfrac{\alpha}{(d_1+\lambda)^2} & & \\ & \ddots & \\ & & \dfrac{1}{d_l+\lambda} + \dfrac{\alpha}{(d_l+\lambda)^2} \end{bmatrix} \mathbf{U}.$$

Ugyanilyen alakra hozva a vizsgálni kívánt $(2\mathbf{B}-\mathbf{BAB})$ mátrixot, a középső diagonális mátrix i-edik eleme a következő lesz:

$$s_i = 2 \cdot \frac{d_i+\lambda+\alpha}{(d_i+\lambda)^2} - \frac{d_i(d_i+\lambda+\alpha)^2}{(d_i+\lambda)^4},$$

ezt átrendezve kapjuk:

$$s_i = \frac{d_i+\lambda+\alpha}{(d_i+\lambda)^4}\left(d_i^2+(3\lambda-\alpha)d_i+2\lambda^2\right).$$

Mivel a d_i értékei nemnegatívok, a λ paraméter pozitív, ezért s_i létezik és véges. Az $\alpha=\lambda$ választás miatt a tört számlálója is pozitív és a szorzat második tényezője is pozitív. Mivel mindegyik s_i érték pozitív, ezért a mátrix is pozitív definit. ezzel állításunkat beláttuk.

4. Numerikus összehasonlítás

Az új algoritmus kidolgozásának elsődleges célja az volt, hogy olyan esetekben lehessen alkalmazni, amikor bonyolult a függvény kiszámítása. Ezért olyan módszerekkel hasonlítottam össze, amelyekben iterációs lépésenként csak egyszer vagy legalábbis kevésszer kell a függvényt kiszámolni. Így az új módszert három algoritmussal hasonlítottam össze: a *Gauss—Newton-*, a *Marquardt-* és egy *módosított*

Gauss—Newton-módszerrel, ahol a módosítás abban áll, hogy az invertálandó mátrix főátlójához egy lépésenként csökkenő λ érték adódik hozzá.

A tesztfeladatokban közös, hogy nem mesterségesen kreáltak, hanem különböző gyakorlati problémák megoldása közben merültek fel. Az algoritmusok összehasonlításának alapja a lépésszám. A tesztfeladatok három csoportba tartoznak. Az elsőben azonos függvényt kellett illeszteni 108 adatsorra, a pontok száma (n) 14 és 17 között változott, a paraméterek száma négy és a függvény a következő alakú

$$f(x) = \frac{a}{1+be^{-cx}} + d,$$

ahol (és a továbbiakban is) az ábécé első betűi jelölik a paramétereket. Ebben az esetben, mivel az egyes adatsorok eléggé hasonlítottak egymásra, nemcsak a lépésszámok átlagát, hanem a szórását is értékelni lehet. Minél kisebb a szórás, annál „stabilabb" az algoritmus erre a feladatra, annál kevésbé érzékeny az adatok kis változására. Az eredmények az 1. táblázatban találhatók. A második feladatcsoportban csak egy adatsor van 225 ponttal, de erre különböző függvényeket kellett illeszteni. Ilyen esetben az illesztés célja nem a függvény paramétereinek a becslése, hanem az extra- és interpoláció. A függvényeket és az eredményeket a 2. táblázat mutatja.

1. TÁBLÁZAT

Módszer	Konvergens	Divergens	Lépésátlag ($\bar{x}$)	Szórás (σ)	$\sigma/\bar{x}$
Gauss—Newton	101	7	4,95	0,89	0,18
Marquardt	108	0	7,02	1,64	0,23
Módosított Marquardt	103	5	5,39	0,70	0,13
Módosított Gauss—Newton	104	4	6,05	0,74	0,12

2. TÁBLÁZAT

Módszer / Függvény	*Gauss—Newton*	*Marquardt*	*Módosított Marquardt*	*Módosított Gauss—Newton*
ax^2+bx+c	2	3	3	3
$\frac{a}{x+b}+c$	25	42	25	*
$-e^{ax+b}+c$	22	22	22	22
$\sqrt{a-bx^2}+c$	*	4	4	4

*: az algoritmus nem konvergált.

A harmadik feladatcsoport az elsőhöz hasonlít, az

$$f(x) = \frac{a}{b+x} + c$$

függvényt kellett 14 adatsorra illeszteni. Az eredményeket a 3. táblázat mutatja (ebben a feladatban a *módosított Gauss—Newton-módszer* olyan rossz eredményeket mutatott, hogy nem tüntettük fel a táblázatban).

Összefoglalva a numerikus eredményeket, azt mondhatjuk, hogy a 3. fejezetben leírt algoritmus általában gyorsabb, mint a *Marquardt-módszer*, és habár lassúbb, mint a *Gauss—Newton-módszer*, de többször konvergál.

3. TÁBLÁZAT

Feladat sorszáma	Pontok száma	Kezdeti fv. érték	Optimum érték	Lépésszám		
				Gauss—Newton	*Marquardt*	*Módosított Marquardt*
1	63	906 390,9	11 566,0	12	*	19
2	72	1 131 098,3	10 586,0	6	19	11
3	78	2 786 584,2	13 452,4	6	44	13
4	78	3 814 861,3	10 337,4	6	22	*
5	78	4 293 941,4	19 835,6	8	24	11
6	85	3 366 081,9	7 293,6	7	*	9
7	86	6 478 400,4	4 226,0	10	*	11
8	101	8 687 468,5	14 455,2	8	*	10
9	108	47 038 093,6	15 986,6	8	13	11
10	108	7 193 160,8	74 996,2	9	*	19
11	109	14 100 817,5	6 410,6	10	20	10
12	111	5 991 438,9	9 555,4	8	22	9
13	112	5 468 925,7	24 157,5	6	22	8
14	126	14 524 798,3	12 716,3	9	11	10

*: az algoritmus nem konvergált.

IRODALOM

[1] BARD, Y., *Nonlinear Parameter Estimation* (Academic Press, New York and London, 1974).
[2] BARD, Y., Comparison of gradient methods for the solution of nonlinear parameter estimation problems, *SIAM J. Numer. Anal.* 7 (1970) 157—186.
[3] DENNIS, J. E. JR., "Some computational techniques for the nonlinear least squares problem", *Numerical Solution of Systems of Nonlinear Algebraic Equations, ed. G. D. Byrne and C. A. Hall*, Academic Press, New York and London, 1973, 157—184.

[4] HARTLEY, H. O. "The modified *Gauss—Newton method* for the fitting of nonlinear regression functions by least squares", *Technometrics* **3** (1961) 269—280.
[5] LEVENBERG, K., "A method for the solution of certain nonlinear problems in least squares", *Quart. App. Math.* **2** (1944) 164—168.
[6] MARQUARDT, D. W., "An algorithm for least squares estimation of nonlinear parameters", *J. Soc. Indust. Appl. Math.* **11** (1963) 431—441.
[7] MEYER, R. R., "Theoretical and computational aspects of non-linear regression", *Nonlinear Programming, ed. J. B. Rosen, O. L. Mangasarian and Ritter,* Academic Press, New York and London, 1970, 465—486.
[8] OSBORNE, M. R., "Some aspects of nonlinear least squares calculations", *Numerical Methods for Nonlinear Optimization, ed. F. A. Lootsma,* Academic Press, New York and London, 1972, 171—190.
[9] PLACKETT, R. L., *Principles of Regression Analysis* (Oxford University Press, Oxford, 1960).
[10] RÓZSA, P., *Lineáris algebra és alkalmazásai* (Műszaki Könyvkiadó, Budapest, 1976).

(Beérkezett: 1980. április 2.)

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

A NEW METHOD FOR SOLVING NONLINEAR CURVERFITTING PROBLEM

T. KUTAS

A new method for the numerical solution of least-squares parameter estimation problem is reported in this paper. The algorithm described in Chapter 3 tries to eliminate the disadvantages of *Gauss—Newton method* similarly to other methods *(Hartley, Marquardt, Meyer)*. The new method can be applied when computation of minimizing function is difficult (e.g. it needs solution of an ordinary differential equation). Numerical results and comparison of several algorithm are described in Chapter 4.

Alkalmazott Matematikai Lapok 6 (1980) 27—63

AZ l_p PROGRAMOZÁSRÓL

TERLAKY TAMÁS

Budapest

Cikkünk az l_p programozási feladatpár tulajdonságait mutatja be, amely az l_p approximációs problémák általánosításaként keletkezett. Az l_p programozás szoros kapcsolatban áll a geometriai programozással, mivel mindkettőben alapvető szerepet játszik a geometriai egyenlőtlenség. Cikkünk felépítése KLAFSZKY EMIL [1] dolgozatának felépítését követi.

Az első részben a matematikai programozásban fontos szerepet játszó dualitási tételeket bizonyítunk. Ezen eredmények többsége E. L. PETERSON és J. G. ECKER [3, 4, 5] cikksorozatában megtalálható. Ezután további tulajdonságokat vizsgálunk, a *Lagrange-függvények* és az optimális megoldások kapcsolatát, regularitást, érzékenységet. Végül a lineáris programozási, az l_p korlátos l_p approximációs, a kvadratikus feltételes kvadratikus programozási, valamint a kompromisszum programozási feladatot mutatjuk be mint az l_p programozási feladatpár speciális eseteit.

1. Bevezetés

Cikkünkben az l_p programozási feladatpárt és annak tulajdonságait mutatjuk be. A bemutatásra kerülő matematikai programozási feladattípus az l_p approximációs problémák általánosításaként keletkezett. Az l_p programozás primál feladatának feltételei az l_p normához hasonló kifejezéseket tartalmaznak, ezért nevezzük l_p norma programozásnak vagy röviden l_p programozásnak a vizsgált feladatpárt.

Rá kell mutatnunk az l_p programozás és a geometriai programozás kapcsolatára. Mindkettőben alapvető szerepet játszik a geometriai egyenlőtlenség. A bizonyítások során felhasznált módszerek és tételek hasonlók, sokszor azonosak is. A bizonyított tételek is hasonlók, de az l_p programozásra bizonyított tételek gyakran erősebbek, mint a geometriai programozásban nekik megfelelő állítások. A szoros kapcsolat ellenére sem igaz az, hogy bármelyik speciális eseteként a másik adódna.

Az l_p programozással először E. L. PETERSON és J. G. ECKER [3, 4, 5] foglalkozott. Cikkünk első pontjaiban (2.—7. pontok) az általuk bizonyított eredményekhez jutunk el. Ennek a résznek a felépítése eltér PETERSON és ECKER cikkeinek felépítésétől, valamint a bizonyítások is lényegesen egyszerűbbek. Ezt az egyszerűsítést az tette lehetővé, hogy ők sem a *Slater-regularitás* fogalmát, sem a *konvex Farkas-tételt* nem használták fel, amelyek segítségével a fent említett módosításokon kívül az l_p programozási feladatpár tulajdonságainak több szempontból való vizsgálatára is lehetőség nyílt.

A cikk további pontjaiban (8.—10. pontok) a feladatpár további tulajdonságait tárgyaljuk (a *Lagrange-függvények* és az optimális megoldáspár kapcsolatát, regularitást, érzékenységet). Ezek a tételek eddig ismeretlenek voltak az l_p programozási feladatpárra, bizonyításuk során a KLAFSZKY EMIL [1] által a geometriai programozás hasonló tételeinek bizonyítására alkalmazott módszereket használtuk fel.

Végül a cikk utolsó pontjában bemutatjuk, hogy miként adódik az l_p programozás speciális eseteként a lineáris programozás (LP), az l_p korlátos l_p approximációs probléma, a kvadratikus feltételes kvadratikus programozás, valamint a kompromisszum programozás.

Bizonyításaink során több, a matematikai programozásban gyakran előforduló tételt használunk fel. Ezeket előzetesen, bizonyítás nélkül közöljük.

1.1. TÉTEL. (*Geometriai egyenlőtlenség*, [1].) Legyenek $\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n \in R$ nem negatív számok. Legyen $\left(\frac{\alpha_i}{0}\right)^0 = 1$, ekkor

$$\left(\frac{\sum_{i=1}^{n} \alpha_i}{\sum_{i=1}^{n} \beta_i}\right)^{\sum_{i=1}^{n} \beta_i} \geqq \prod_{i=1}^{n} \left(\frac{\alpha_i}{\beta_i}\right)^{\beta_i}$$

egyenlőséggel akkor és csak akkor áll fenn, ha

$$\alpha_i \sum_{i=1}^{n} \beta_i = \beta_i \sum_{i=1}^{n} \alpha_i, \quad i = 1, \ldots, n.$$

Felhasználásra kerül ennek az egyenlőtlenségnek egy speciális esete is.

Legyen $\alpha, \beta \in R$ és $p > 1$, $q > 1$ számok úgy, hogy $\frac{1}{p} + \frac{1}{q} = 1$. Ekkor

$$\alpha\beta \leqq \frac{1}{p}|\alpha|^p + \frac{1}{q}|\beta|^q.$$

Egyenlőség akkor és csak akkor áll fenn, ha

$$|\alpha|^{p-1} \operatorname{sgn} \alpha = \beta,$$

vagy ami ezzel ekvivalens

$$|\beta|^{q-1} \operatorname{sgn} \beta = \alpha.$$

1.2. KONVEX FARKAS-TÉTEL ([6]). Mielőtt a tételt kimondanánk, szükséges ismertetnünk a *Slater-regularitás* fogalmát: Legyenek a $g_1, \ldots, g_k$ függvények konvexek a $C \subset R^m$ konvex halmazon. Ez a függvényrendszer kielégíti a *Slater-regularitási feltételt*, ha létezik $\mathbf{x}^0$ relatív belső pontja C-nek úgy, hogy

$g_j(\mathbf{x}^0) \leqq 0$ azon j-k esetén, mikor g_j lineáris,

$g_j(\mathbf{x}^0) < 0$ azon j-k esetén, mikor g_j nem lineáris.

Így most már kimondhatjuk a tételt:

Legyenek $g_1, \ldots, g_k, f$ konvex függvények a $C \subset R^m$ konvex halmazon. Tegyük fel, hogy a $g_1, \ldots, g_k$ függvények kielégítik a *Slater-regularitási feltételt*, ekkor ha a

$$g_j(\mathbf{x}) \leqq 0, \quad j = 1, \ldots, k,$$

$$f(\mathbf{x}) < 0,$$

$$\mathbf{x} \in C$$

rendszernek nincs megoldása, akkor van olyan $\mathbf{y}=(\eta_1, \ldots, \eta_k)\geqq\mathbf{0}$ vektor, hogy

$$f(\mathbf{x})+\sum_{j=1}^{k}\eta_j g_j(\mathbf{x})\geqq 0 \quad \text{minden } \mathbf{x}\in C \text{ esetén.}$$

1.3. KUHN—TUCKER-TÉTEL ([6]). Legyenek a $g_1, \ldots, g_k, f$ függvények konvexek a $C\subset R^m$ konvex halmazon. Tegyük fel, hogy a $g_1, \ldots, g_k$ függvényrendszer kielégíti a *Slater-regularitási feltételt*. Ekkor $\mathbf{x}^*\in C$ optimális megoldása a

$$g_j(\mathbf{x})\leqq 0, \quad j=1, \ldots, k$$

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

konvex programozási feladatnak akkor és csak akkor, ha létezik $\mathbf{t}^*=(\tau_1^*, \ldots, \tau_k^*)\geqq\mathbf{0}$ vektor úgy, hogy $(\mathbf{x}^*, \mathbf{t}^*)$ nyeregpontja az

$$L(\mathbf{x}, \mathbf{t})=f(\mathbf{x})+\sum_{j=1}^{n}\tau_j g_j(\mathbf{x})$$

Lagrange-függvénynek $C\times R_+^m$-on.

1.4. TUCKER-TÉTEL ([7]). Legyenek $\mathbf{y}\in R^m$, $\mathbf{x}_1\in R^{n_1}$, $\mathbf{x}_2\in R^{n_2}$, és $\mathbf{C}_1: n_1\times m$-es, $\mathbf{C}_2: n_2\times m$-es tetszőleges mátrixok. Tekintsük a

$$\mathbf{C}_1\mathbf{y}\geqq\mathbf{0}$$

$$\mathbf{C}_2\mathbf{y}=\mathbf{0}$$

egyenlőtlenség-rendszert, és ehhez rendeljük hozzá a következő egyenlőtlenség-rendszert:

$$\mathbf{x}_1\mathbf{C}_1+\mathbf{x}_2\mathbf{C}_2=\mathbf{0}$$

$$\mathbf{x}_1\geqq\mathbf{0}$$

Ennek az egyenlőtlenségrendszer-párnak léteznek olyan $\mathbf{y}', \mathbf{x}_1', \mathbf{x}_2'$ megoldásai, hogy

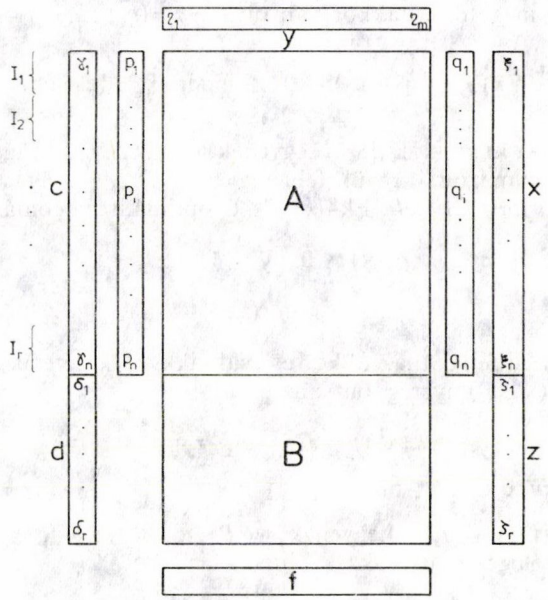
$$\mathbf{C}_1\mathbf{y}'+\mathbf{x}_1'>\mathbf{0}.$$

Az ilyen megoldásokat nevezzük *Tucker-megoldásnak*.

2. Az l_p programozási feladatpár és alapvető lemmája

Ebben a fejezetben az l_p programozási feladatpárt definiáljuk, és egy egyenlőtlenség belátása után az l_p programozás fő lemmáját bizonyítjuk.

Legyen $\mathbf{A}: n\times m$, $\mathbf{B}: r\times m$-es mátrix, valamint $\mathbf{y}, \mathbf{f}\in R^m$, $\mathbf{c}, \mathbf{x}\in R^n$, $\mathbf{d}, \mathbf{z}\in R^r$ és $p_i>1$, $q_i>1$ számok úgy, hogy $\frac{1}{p_i}+\frac{1}{q_i}=1$, $i=1, \ldots, n$, továbbá I_k, $k=1, \ldots, r$ indexhalmazok úgy, hogy $I_k\cap I_j=\varnothing$, $j\neq k$ és $\bigcup_{k=1}^{r} I_k=\{1, \ldots, n\}$.



1. ábra

Az l_p programozás primál feladata:

$$G_k(\mathbf{y}) = \sum_{i \in I_k} \frac{1}{p_i} |\mathbf{a}_i \mathbf{y} - \gamma_i|^{p_i} + \mathbf{b}_k \mathbf{y} - \delta_k \leqq 0, \quad k = 1, \ldots, r$$

$$\sup \mathbf{fy}$$

Az l_p programozás duál feladata:

$$\mathbf{xA} + \mathbf{zB} = \mathbf{f},$$

$$\zeta_k = 0 \Rightarrow \xi_i = 0, \quad i \in I_k, \quad k = 1, \ldots, r,$$

$$\inf \left\{ \mathbf{cx} + \mathbf{dz} + \sum_{\substack{k=1 \\ \zeta_k > 0}}^{r} \zeta_k \sum_{i \in I_k} \frac{1}{q_i} \left| \frac{\xi_i}{\zeta_k} \right|^{q_i} \right\}.$$

Vezessük be a következő jelöléseket:

$$F_k(\mathbf{x}, \mathbf{z}) = \begin{cases} \zeta_k \sum_{i \in I_k} \frac{1}{q_i} \left| \frac{\xi_i}{\zeta_k} \right|^{q_i}, & \text{ha} \quad \zeta_k > 0, \\ 0, & \text{ha} \quad \zeta_k = 0, \end{cases} \qquad k = 1, \ldots, r,$$

$$F(\mathbf{x}, \mathbf{z}) = \sum_{k=1}^{r} F_k(\mathbf{x}, \mathbf{z}).$$

Így a duál feladat célfüggvénye: $\mathbf{cx}+\mathbf{dz}+F(\mathbf{x}, \mathbf{z})$. Továbbá legyen:

$$P = \{\mathbf{y} | G_k(\mathbf{y}) \leqq 0,\ k = 1, \ldots, r\},$$

$$P^* = \{\mathbf{y} | \mathbf{y} \in P \text{ és } \mathbf{y} \text{ primál optimális}\},$$

$$D = \{(\mathbf{x}, \mathbf{z}) | \mathbf{xA}+\mathbf{zB} = \mathbf{f},\ \mathbf{z} \geqq 0,\ \zeta_k = 0 \Rightarrow \xi_i = 0,\ i \in I_k,\ k = 1, \ldots, r\},$$

$$D^* = \{(\mathbf{x}, \mathbf{z}) | (\mathbf{x}, \mathbf{z}) \in D \text{ és } (\mathbf{x}, \mathbf{z}) \text{ duál optimális}\},$$

$$\mu = \sup_{\mathbf{y} \in P} \mathbf{fy}, \quad \nu = \inf_{(\mathbf{x}, \mathbf{z}) \in D} \{\mathbf{cx}+\mathbf{dz}+F(\mathbf{x}, \mathbf{z})\}.$$

Most egy egyenlőtlenséget bizonyítunk, mely a geometriai egyenlőtlenséggel együtt fontos eszköze lesz vizsgálatainknak.

2.1. LEMMA. Legyen $q>1$, $\alpha, \bar{\alpha} \geqq 0$, $\beta, \bar{\beta} \in R$, $\alpha=0 \Rightarrow \beta=0$, $\bar{\alpha}=0 \Rightarrow \bar{\beta}=0$, és $\alpha+\bar{\alpha}>0$, ekkor fennáll a következő egyenlőtlenség:

$$(\alpha+\bar{\alpha})^{1-q}|\beta+\bar{\beta}|^q \leqq \alpha^{1-q}|\beta|^q+\bar{\alpha}^{1-q}|\bar{\beta}|^q.$$

Bizonyítás. Ha $|\beta+\bar{\beta}|=0$, vagy ha $\alpha=0$, vagy $\bar{\alpha}=0$, akkor az egyenlőtlenség triviálisan fennáll. Nyilván szimmetrikus α és $\bar{\alpha}$, valamint β és $\bar{\beta}$ szerepe.

A továbbiakban feltesszük, hogy $\alpha>0$, $\bar{\alpha}>0$ és $|\beta+\bar{\beta}| \neq 0$.

Ha $\beta, \bar{\beta} \geqq 0$, akkor az alábbi egyenlőtlenséget kell bizonyítani: $(\alpha+\bar{\alpha})^{1-q}(\beta+\bar{\beta})^q \leqq \alpha^{1-q}\beta^q+\bar{\alpha}^{1-q}\bar{\beta}^q$. Ez fennáll, mivel a geometriai egyenlőtlenséget alkalmazva az $A \in (0, 1)$, $B \in [0, 1)$ értékekkel nyerjük:

$$1 = 1^{q-1} \leqq \left[\left(\frac{B}{A}\right)^B \left(\frac{1-B}{1-A}\right)^{1-B}\right]^{q-1} \leqq B\left(\frac{B}{A}\right)^{q-1} + (1-B)\left(\frac{1-B}{1-A}\right)^{q-1}.$$

Átalakítva:

$$1 \leqq A^{1-q}B^q + (1-A)^{1-q}(1-B)^q.$$

Legyen $A = \frac{\alpha}{\alpha+\bar{\alpha}}$ és $B = \frac{\beta}{\beta+\bar{\beta}}$, ekkor $1-A = \frac{\bar{\alpha}}{\alpha+\bar{\alpha}}$ és $1-B = \frac{\bar{\beta}}{\beta+\bar{\beta}}$. Behelyettesítve kapjuk a kívánt egyenlőtlenséget:

$$(\alpha+\bar{\alpha})^{1-q}(\beta+\bar{\beta})^q \leqq \alpha^{1-q}\beta^q + \bar{\alpha}^{1-q}\bar{\beta}^q.$$

Ha $\beta \geqq 0$ és $\bar{\beta}<0$, ekkor $(\alpha+\bar{\alpha})^{1-q}|\beta+\bar{\beta}|^q \leqq (\alpha+\bar{\alpha})^{1-q}(\beta-\bar{\beta})^q \leqq \alpha^{1-q}\beta^q + \bar{\alpha}^{1-q}(-\bar{\beta})^q = \alpha^{1-q}|\beta|^q + \bar{\alpha}^{1-q}|\bar{\beta}|^q$.

Ha $\beta<0$ és $\bar{\beta}<0$, akkor $(\alpha+\bar{\alpha})^{1-q}|\beta+\bar{\beta}|^q = (\alpha+\bar{\alpha})^{1-q}(-\beta-\bar{\beta})^q \leqq \alpha^{1-q}(-\beta)^q + \bar{\alpha}^{1-q}(-\bar{\beta})^q = \alpha^{1-q}|\beta|^q + \bar{\alpha}^{1-q}|\bar{\beta}|^q$.

Így beláttuk az egyenlőtlenséget.

2.2. LEMMA. *(Az l_p programozás fő lemmája.)* Legyen $\mathbf{y} \in P$ és $(\mathbf{x}, \mathbf{z}) \in D$, akkor $\mathbf{fy} \leqq \mathbf{cx}+\mathbf{dz}+F(\mathbf{x}, \mathbf{z})$ egyenlőséggel akkor és csak akkor, ha

$$\zeta_k G_k(\mathbf{y}) = 0, \quad k = 1, \ldots, r$$

$$\zeta_k \operatorname{sgn}(\mathbf{a}_i\mathbf{y}-\gamma_i)|\mathbf{a}_i\mathbf{y}-\gamma_i|^{p_i-1} = \xi_i, \quad i \in I_k, \quad k = 1, \ldots, r,$$

vagy ami ezzel ekvivalens,

$$\zeta_k G_k(\mathbf{y}) = 0, \quad k = 1, \ldots, r,$$

$$\zeta_k = 0, \quad \text{vagy} \quad \mathbf{a}_i\mathbf{y} - \gamma_i = \operatorname{sgn} \xi_i \left|\frac{\xi_i}{\zeta_k}\right|^{q_i - 1}, \quad i \in I_k, \quad k = 1, \ldots, r.$$

Bizonyítás. Előbb a duál, majd a primál feltételt használjuk fel:

$$\mathbf{fy} = \mathbf{xAy} + \mathbf{zBy} \leqq \mathbf{xAy} + \mathbf{dz} - \sum_{k=1}^{r} \zeta_k \sum_{i \in I_k} \frac{1}{p_i} |\mathbf{a}_i\mathbf{y} - \gamma_i|^{p_i}.$$

Egyenlőség akkor és csak akkor áll fenn, ha $\zeta_k G_k(\mathbf{y}) = 0$, $k = 1, \ldots, r$. Alkalmazzuk a speciális geometriai egyenlőtlenséget a következő esetben: $p = p_i$, $q = q_i$, $\alpha = \mathbf{a}_i\mathbf{y} - \gamma_i$,

$$\beta = \begin{cases} \dfrac{\xi_i}{\zeta_k}, & \text{ha} \quad \zeta_k > 0 \\ 0, & \text{ha} \quad \zeta_k = 0 \end{cases}.$$

Egyenlőség fog fenn állni akkor és csak akkor, ha

$$\zeta_k \operatorname{sgn}(\mathbf{a}_i\mathbf{y} - \gamma_i) |\mathbf{a}_i\mathbf{y} - \gamma_i|^{p_i - 1} = \xi_i, \quad i \in I_k, \quad k = 1, \ldots, r,$$

vagy ami ezzel ekvivalens

$$\zeta_k = 0, \quad \text{vagy} \quad \mathbf{a}_i\mathbf{y} - \gamma_i = \operatorname{sgn} \xi_i \left|\frac{\xi_i}{\zeta_k}\right|^{q_i - 1}, \quad i \in I_k, \quad k = 1, \ldots, r.$$

Így nyertük, hogy

$$\mathbf{fy} \leqq \mathbf{xAy} + \mathbf{dz} + \sum_{\substack{k=0 \\ \zeta_k > 0}}^{r} \zeta_k \sum_{i \in I_k} \frac{1}{q_i} \left|\frac{\xi_i}{\zeta_k}\right|^{q_i} - \left[\sum_{k=1}^{r} \sum_{i \in I_k} \xi_i \mathbf{a}_i\mathbf{y} - \mathbf{cx}\right] =$$

$$= \mathbf{xAy} + \mathbf{dz} + F(\mathbf{x}, \mathbf{z}) - \mathbf{xAy} + \mathbf{cx} = \mathbf{cx} + \mathbf{dz} + F(\mathbf{x}, \mathbf{z}).$$

Tehát állításunkat beláttuk.

KÖVETKEZMÉNY. *a)* Ha $P \neq \varnothing$, akkor a duál feladat célfüggvénye alulról korlátos, *b)* Ha $D \neq \varnothing$, akkor a primál feladat célfüggvénye felülről korlátos.

Bizonyítás. Az állítás a fő lemmából nyilvánvalóan következik.

3. Elemi tulajdonságok

Az l_p programozási feladatpár elemi tulajdonságait vizsgálva megmutatjuk, hogy a G_k és F függvények konvexek, majd a P és D halmazok tulajdonságait vizsgáljuk.

A G_k függvények konvexek, mivel az abszolút érték függvény konvex, és konvex függvények összege is konvex. Tehát így a primál feladat egy konvex programozási feladat.

3.1. LEMMA. Legyen $C=\{(\mathbf{x}, \mathbf{z})|\mathbf{z}\geqq\mathbf{0}, \zeta_k=0\Rightarrow\xi_i=0,\ i\in I_k,\ \forall k\}$, akkor F-re a következő három állítás igaz ezen a konvex halmazon:

a) $F(\mathbf{x}, \mathbf{z})\geqq 0$,

b) $F(\lambda\mathbf{x}, \lambda\mathbf{z})=\lambda F(\mathbf{x}, \mathbf{z}),\quad \lambda\geqq 0$,

c) $F(\mathbf{x}+\bar{\mathbf{x}}, \mathbf{z}+\bar{\mathbf{z}})\leqq F(\mathbf{x}, \mathbf{z})+F(\bar{\mathbf{x}}, \bar{\mathbf{z}})$.

Vagyis F nem negatív, pozitív homogén és szubadditív, s így konvex függvény C-n.

Bizonyítás. a) és *b)* triviális állítás. A *c)* állítás bizonyítása:

$$F(\mathbf{x}+\bar{\mathbf{x}}, \mathbf{z}+\bar{\mathbf{z}}) = \sum_{\substack{k=1\\ \zeta_k+\bar{\zeta}_k>0}}^{r} (\zeta_k+\bar{\zeta}_k) \sum_{i\in I_k} \frac{1}{q_i}\left|\frac{\xi_i+\bar{\xi}_i}{\zeta_k+\bar{\zeta}_k}\right|^{q_i} =$$

$$= \sum_{\substack{k=1\\ \zeta_k+\bar{\zeta}_k>0}}^{r} \sum_{i\in I_k} \frac{1}{q_i}(\zeta_k+\bar{\zeta}_k)^{1-q_i}|\xi_i+\bar{\xi}_i|^{q_i} \leqq$$

$$\leqq \sum_{\substack{k=1\\ \zeta_k>0}}^{r} \zeta_k \sum_{i\in I_k} \frac{1}{q_i}\left|\frac{\xi_i}{\zeta_k}\right|^{q_i} + \sum_{\substack{k=1\\ \bar{\zeta}_k>0}}^{r} \bar{\zeta}_k \sum_{i\in I_k} \frac{1}{q_i}\left|\frac{\bar{\xi}_i}{\bar{\zeta}_k}\right|^{q_i} = F(\mathbf{x}, \mathbf{z})+F(\bar{\mathbf{x}}, \bar{\mathbf{z}}).$$

Itt a 2.1. lemma egyenlőtlenségét alkalmaztuk $\alpha=\zeta_k$, $\bar{\alpha}=\bar{\zeta}_k$, $\beta=\xi_i$ és $\bar{\beta}=\bar{\xi}_i$ mellett. Így állításunkat beláttuk.

3.2. LEMMA. A primál feltételi halmazt a következő két konvex poliéder fogja közre:

a) $P\subset\{\mathbf{y}|\mathbf{B}\mathbf{y}\leqq\mathbf{d}\}$,

b) $P\supset\{\mathbf{y}|\mathbf{A}\mathbf{y}=\mathbf{c},\ \mathbf{B}\mathbf{y}\leqq\mathbf{d}\}$.

Bizonyítás. Mindkét állítás triviális.

3.3. LEMMA. Legyen $\tilde{\mathbf{y}}\in R^m$ olyan, hogy $\mathbf{A}\tilde{\mathbf{y}}=\mathbf{0}$ és $\mathbf{B}\tilde{\mathbf{y}}\leqq\mathbf{0}$, akkor $\mathbf{y}\in P$ és tetszőleges $\vartheta\geqq 0$ esetén $\mathbf{y}+\vartheta\tilde{\mathbf{y}}\in P$.

Bizonyítás. Legyen $\mathbf{y}\in P$, akkor

$$0 \geqq \sum_{i\in I_k} \frac{1}{p_i}|\mathbf{a}_i\mathbf{y}-\gamma_i|^{p_i}+\mathbf{b}_k\mathbf{y}-\delta_k \geqq$$

$$\geqq \sum_{i\in I_k} \frac{1}{p_i}|\mathbf{a}_i(\mathbf{y}+\vartheta\tilde{\mathbf{y}})-\gamma_i|^{p_i}+\mathbf{b}_k(\mathbf{y}+\vartheta\tilde{\mathbf{y}})-\delta_k,\quad k=1, \ldots, r.$$

Tehát $\mathbf{y}+\vartheta\tilde{\mathbf{y}}\in P$.

3.4. LEMMA Legyen $P\neq\varnothing$, ebben az esetben P akkor és csak akkor korlátos, ha $\mathbf{B}\mathbf{y}\leqq\mathbf{0}$, $\mathbf{A}\mathbf{y}=\mathbf{0}$-nak csak triviális megoldása létezik.

Bizonyítás. Ha P korlátos, akkor a 3.3. lemma miatt $\mathbf{B}\mathbf{y}\leqq\mathbf{0}$, $\mathbf{A}\mathbf{y}=\mathbf{0}$-nak csak triviális megoldása létezik.

Ha P nem korlátos, legyen $\mathbf{y}\in P$, ekkor ehhez létezik $\mathbf{y}^0\neq 0$ úgy, hogy $\mathbf{y}+\vartheta\mathbf{y}^0\in P$ minden $\vartheta\geqq 0$ esetén. Mivel $P\subset\{\mathbf{y}|\mathbf{B}\mathbf{y}\leqq\mathbf{d}\}$, így $\mathbf{B}(\mathbf{y}+\vartheta\mathbf{y}^0)\leqq\mathbf{d}$ minden $\vartheta\geqq 0$ esetén, tehát $\mathbf{B}\mathbf{y}^0\leqq\mathbf{0}$ kell hogy teljesüljön.

Írjuk fel a primál feladat feltételét az $\mathbf{y}+\vartheta\mathbf{y}^0\in P$ pontra.

$$\sum_{i\in I_k}\frac{1}{p_i}|\mathbf{a}_i(\mathbf{y}+\vartheta\mathbf{y}^0)-\gamma_i|^{p_i}+\mathbf{b}_k(\mathbf{y}+\vartheta\mathbf{y}^0)-\delta_k\leqq 0,\quad k=1,\ldots,r.$$

Ebből következik, hogy az alábbi egyenlőtlenség is igaz.

$$\frac{1}{p_i}|\mathbf{a}_i(\mathbf{y}+\vartheta\mathbf{y}^0)-\gamma_i|^{p_i}\leqq\delta_k-\mathbf{b}_k(\mathbf{y}+\vartheta\mathbf{y}^0),\quad i\in I_k,\quad k=1,\ldots,r.$$

Mivel $\delta_k-\mathbf{b}_k(\mathbf{y}+\vartheta\mathbf{y}^0)\geqq 0$, elvégezhető az $\frac{1}{p_i}$-edik hatványra emelés.

$$-\left[p_i(\delta_k-\mathbf{b}_k(\mathbf{y}+\vartheta\mathbf{y}^0))\right]^{\frac{1}{p_i}}\leqq\mathbf{a}_i\mathbf{y}-\gamma_i+\vartheta\mathbf{a}_i\mathbf{y}^0\leqq\left[p_i(\delta_k-\mathbf{b}_k(\mathbf{y}+\vartheta\mathbf{y}^0))\right]^{\frac{1}{p_i}},\ i\in I_k,\ k=1,\ldots,r.$$

Ezt átrendezve

$$\frac{\gamma_i-\mathbf{a}_i\mathbf{y}}{\vartheta}-\frac{p_i(\delta_k-\mathbf{b}_k(\mathbf{y}+\vartheta\mathbf{y}^0))^{\frac{1}{p_i}}}{\vartheta}\leqq\mathbf{a}_i\mathbf{y}^0\leqq$$

$$\leqq\frac{\gamma_i-\mathbf{a}_i\mathbf{y}}{\vartheta}+\frac{\left[p_i(\delta_k-\mathbf{b}_k(\mathbf{y}+\vartheta\mathbf{y}^0))\right]^{\frac{1}{p_i}}}{\vartheta},\quad i\in I_k,\quad k=1,\ldots,r.$$

Ez az egyenlőtlenség minden $\vartheta\geqq 0$-ra fennáll, és $\vartheta\to+\infty$ esetén $p_i>1$ miatt $\frac{\gamma_i-\mathbf{a}_i\mathbf{y}}{\vartheta}\to 0$, valamint $\frac{\left[p_i(\delta_k-\mathbf{b}_k(\mathbf{y}+\vartheta\mathbf{y}^0))\right]^{\frac{1}{p_i}}}{\vartheta}\to 0$, így $\mathbf{a}_i\mathbf{y}^0=0$, $i\in I_k$, $k=1,\ldots,r$.

Tehát $\mathbf{B}\mathbf{y}^0\leqq 0$, $\mathbf{A}\mathbf{y}^0=\mathbf{0}$, $\mathbf{y}^0\neq\mathbf{0}$, így ha P nem korlátos, akkor a $\mathbf{B}\mathbf{y}\leqq\mathbf{0}$, $\mathbf{A}\mathbf{y}=\mathbf{0}$ rendszernek van nem triviális megoldása.

A 3.5. lemma és következményei a redukált feladatpár definiálásához szükségesek, valamint annak vizsgálatakor játszanak fontos szerepet.

3.5. LEMMA. *a)* D konvex,

b) $D=D+D(f=0)$,

ahol $D(f=0)=\{(\mathbf{x},\mathbf{z})|\mathbf{x}\mathbf{A}+\mathbf{z}\mathbf{B}=\mathbf{0},\ \mathbf{z}\geqq\mathbf{0},\ \xi_k=0\Rightarrow\xi_i=0,\ i\in I_k,\ \forall k\}$.

Bizonyítás. Mindkét állítás könnyen ellenőrizhető.

Megjegyzés: Mivel F pozitív homogén és szubadditív függvény a $C\supset D$ konvex halmazon a 2.1. lemma miatt és D konvex, így az l_p programozás duál feladata is egy konvex programozási feladat.

1. KÖVETKEZMÉNY. Ha minden k-hoz $k=1,\ldots,r$ van $(\mathbf{x}^k,\mathbf{z}^k)\in D$ ugy, hogy $\zeta_k^k>0$, akkor létezik olyan $(\mathbf{x},\mathbf{z})\in D$, hogy $\mathbf{z}>\mathbf{0}$.

Bizonyítás. Az $(\mathbf{x}^k,\mathbf{z}^k)$, $k=1,\ldots,r$ vektorok pozitív együtthatós konvex lineáris kombinációja D-ben marad a lemma miatt, továbbá erre $\mathbf{z}>\mathbf{0}$.

Vezessük be a következő jelölést:

$$D_0 = \{(\mathbf{x}, \mathbf{z}, \zeta_0) | \mathbf{x}\mathbf{A} + \mathbf{z}\mathbf{B} = \zeta_0 \mathbf{f},\ \mathbf{z}, \zeta_0 \geqq 0,\ \zeta_k = 0 \Rightarrow \xi_i = 0,\ i \in I_k,\ \forall k\}.$$

Nyilvánvaló az, hogy tetszőleges $(\mathbf{x}, \mathbf{z}, \zeta_0) \in D_0$ esetén, ha $\zeta_0 > 0$, akkor $\left(\frac{\mathbf{x}}{\zeta_0}, \frac{\mathbf{z}}{\zeta_0}\right) \in D$ és ha $\zeta_0 = 0$, akkor $(\mathbf{x}, \mathbf{z}) \in D(f=0)$.

3.1. DEFINÍCIÓ. Legyen adott D_0, ekkor osszuk a $\{0, 1, \ldots, r\}$ halmazt két részhalmazra. Legyen $\Gamma = \{k | 0 \leqq k \leqq r,\ \zeta_k > 0$ legalább egy $(\mathbf{x}, \mathbf{z}, \zeta_0) \in D_0$ esetén$\}$, $\bar{\Gamma} = \{k | 0 \leqq k \leqq r,\ \zeta_k = 0$ minden $(\mathbf{x}, \mathbf{z}, \zeta_0) \in D_0$ esetén$\}$.

Nyilván igaz az, hogy $\Gamma \cap \bar{\Gamma} = \varnothing$ és $\Gamma \cup \bar{\Gamma} = \{0, 1, \ldots, r\}$.

2. KÖVETKEZMÉNY. Létezik olyan $(\mathbf{x}^+, \mathbf{z}^+, \zeta_0^+) \in D_0$, hogy $\zeta_k^+ > 0$ $k \in \Gamma$ és tetszőleges $(\mathbf{x}, \mathbf{z}, \zeta_0) \in D_0$ esetén $\zeta_k = \xi_i = 0$ $i \in I_k$, $k \in \bar{\Gamma}$.

Bizonyítás. Az állítás nyilvánvaló, mivel D_0 konvex kúp.

Vegyük észre azt, hogy ha $0 \in \bar{\Gamma}$, akkor $D = \varnothing$ és ha $0 \in \Gamma$, akkor $D \neq \varnothing$. Ebben az esetben legyen $\Gamma_0 = \Gamma - \{0\}$, amire ekkor igaz, hogy $\Gamma_0 \cap \bar{\Gamma} = \varnothing$ és $\{1, \ldots, r\} = \Gamma_0 \cup \bar{\Gamma}$.

4. Slater-reguláris primál feladatok

Mivel a primál feladat feltételi függvényei konvexek, kézenfekvő a *konvex Farkas-tétel* alkalmazásával vizsgálni a feladatpár tulajdonságait. Ehhez szükséges, hogy a primál feladat kielégítse a *Slater-regularitási feltételt,* amely esetünkben a következő: létezik $\bar{\mathbf{y}} \in P$ úgy, hogy $G_k(\bar{\mathbf{y}}) < 0$ minden olyan k esetén, amikor $I_k \neq \varnothing$.

4.1. TÉTEL. Ha a primál feladat *Slater-reguláris* és μ véges, akkor létezik $(\mathbf{x}^*, \mathbf{z}^*) \in D$ úgy, hogy

$$\mathbf{c}\mathbf{x}^* + \mathbf{d}\mathbf{z}^* + F(\mathbf{x}^*, \mathbf{z}^*) = \mu.$$

Bizonyítás. Tekintsük a következő feltételrendszert:

$$\sum_{i \in I_k} \frac{1}{p_i} |\pi_i|^{p_i} + \mathbf{b}_k \mathbf{y} - \delta_k \leqq 0, \quad k = 1, \ldots, r, \tag{4.1}$$

$$a_i \mathbf{y} - \gamma_i = \pi_i,\ i \in I_k, \quad k = 1, \ldots, r.$$

A (4.1) feltételrendszer kielégíti a *Slater regularitási feltételt* és ennek megoldáshalmazán $\mu \geqq \mathbf{f}\mathbf{y}$. Így

$$\mu - \mathbf{f}\mathbf{y} < 0 \tag{4.2}$$

nem áll fenn egyetlen megoldására sem (4.1)-nek.

A *konvex Farkas-tételt* alkalmazva (4.1) és (4.2)-re nyerjük, hogy létezik $\mathbf{z}^* = (\zeta_1^*, \ldots, \zeta_r^*) \geqq 0$ és $\mathbf{x}^* = (\xi_1^*, \ldots, \xi_n^*)$ vektor úgy, hogy tetszőleges $\mathbf{y}$ és $\pi_1, \ldots, \pi_n$ esetén

$$\mu - \mathbf{f}\mathbf{y} + z^* \mathbf{B}\mathbf{y} - \mathbf{d}\mathbf{z}^* + \sum_{i=1}^{n} \xi_i^* (\mathbf{a}_i \mathbf{y} - \gamma_i - \pi_i) + \sum_{k=1}^{r} \zeta_k^* \sum_{i \in I_k} \frac{1}{p_i} |\pi_i|^{p_i} \geqq 0,$$

vagy rendezve:

$$\mu+(\mathbf{z}^*\mathbf{B}+\mathbf{x}^*\mathbf{A}-\mathbf{f})\mathbf{y}-\mathbf{c}\mathbf{x}^*-\mathbf{d}\mathbf{z}^*-\sum_{i=1}^{n}\xi_i^*\pi_i+\sum_{k=1}^{r}\zeta_k^*\sum_{i\in I_k}\frac{1}{p_i}|\pi_i|^{p_i}\geqq 0.$$

Mivel ez az egyenlőtlenség minden $\pi_1, \ldots, \pi_n$ esetén fennáll, ezért $\zeta_k^*=0\Rightarrow\xi_i^*=0$, $i\in I_k$, $k=1, \ldots, r$, mert különben elég nagy π_i-t választva ellentmondásra jutnánk. Hasonlóan az egyenlőtlenség minden $\mathbf{y}$-ra is igaz, így $\mathbf{z}^*\mathbf{B}+\mathbf{x}^*\mathbf{A}=\mathbf{f}$, mert különben választható lenne olyan $\mathbf{y}$, hogy a bal oldal negatív legyen. Tehát $(\mathbf{x}^*, \mathbf{z}^*)\in D$.

Az egyenlőtlenség bal oldalához $F(\mathbf{x}^*, \mathbf{z}^*)$-ot hozzáadva és ki is vonva:

$$\mu-\mathbf{c}\mathbf{x}^*-\mathbf{d}\mathbf{z}^*-F(\mathbf{x}^*, \mathbf{z}^*)+\sum_{\substack{k=1\\ \zeta_k^*>0}}^{r}\zeta_k^*\left[\sum_{i\in I_k}\left(\frac{1}{p_i}|\pi_i|^{p_i}+\frac{1}{q_i}\left|\frac{\xi_i^*}{\zeta_k^*}\right|^{q_i}-\frac{\xi_i^*}{\zeta_k^*}\pi_i\right)\right]\geqq 0.$$

Mivel az egyenlőtlenség minden $\pi_1, \ldots, \pi_n$-re fennáll, ezért választhatjuk a $\pi_i=\operatorname{sgn}\xi_i^*\left|\frac{\xi_i^*}{\zeta_k^*}\right|^{q_i-1}$, $i\in I_k$, $k=1, \ldots, r$ értékeket, ekkor a zárójelben álló kifejezés értéke a speciális geometriai egyenlőtlenség egyensúlyi feltétele miatt nulla.

Így $\mu\geqq\mathbf{c}\mathbf{x}^*+\mathbf{d}\mathbf{z}^*+F(\mathbf{x}^*, \mathbf{z}^*)$, de $(\mathbf{x}^*, \mathbf{z}^*)\in D$-ből a fő lemma miatt a fordított egyenlőtlenség is igaz, így

$$\mu=\mathbf{c}\mathbf{x}^*+\mathbf{d}\mathbf{z}^*+F(\mathbf{x}^*, \mathbf{z}^*).$$

Tehát ekkor létezik optimális megoldása a duál feladatnak, valamint a primál és a duál feladatok optimális értékei megegyeznek.

KÖVETKEZMÉNY. Ha a primál feladat *Slater-reguláris*, akkor μ akkor és csak akkor véges, ha $D^*\neq\varnothing$ és ekkor $\mu=v$.

Bizonyítás. Állításunk a fő lemmából és a 3.1. tételből következik.

5. Slater-reguláris duál feladatok

A *Slater-reguláris* duál feladatok megkülönböztetett szerepet játszanak az l_p programozás elméletében és alkalmazásaiban. Mivel az l_p programozási duál feladat is egy konvex programozási feladat, így ennek vizsgálatában is kézenfekvő a *konvex Farkas-tétel* alkalmazása.

Az l_p programozás duál feladata esetén a *Slater-regularitás* definiálása nem olyan egyszerű, mint a primál feladatnál, mivel itt logikai feltételek is szerepelnek. Így ebben az esetben a definíció a következő. Legyen $C=\{(\mathbf{x}, \mathbf{z})|\mathbf{z}\geqq\mathbf{0}, \zeta_k=0\Rightarrow\xi_i=0, i\in I_k, k=1, \ldots, r\}$. C nyilván konvex halmaz. A feltételi függvények lineárisak, így a duál feladat kielégíti a *Slater regularitási feltételt,* ha létezik olyan $(\mathbf{x}, \mathbf{z})\in D$, hogy $\mathbf{z}>\mathbf{0}$. (Mivel pontosan az ilyen $(\mathbf{x}, \mathbf{z})$ pontok a relatív belső pontjai C-nek.)

Ez azt jelenti, hogy a 3.1. definícióban bevezetett Γ indexhalmazra igaz, hogy $\Gamma=\{0, 1, \ldots, r\}$.

5.1. LEMMA. Legyen $\mathbf{f}=\mathbf{0}$, a duál feladat *Slater-reguláris* és minden $(\mathbf{x}, \mathbf{z})\in D$ esetén $\mathbf{c}\mathbf{x}+\mathbf{d}\mathbf{z}+F(\mathbf{x}, \mathbf{z})\geqq 0$. Ekkor van olyan $\bar{\mathbf{y}}\in R^m$, hogy $G_k(\bar{\mathbf{y}})\leqq 0$, $k=1, \ldots, r$ $(P\neq\varnothing)$.

Bizonyítás. Tekintsük a következő rendszert:

$$\mathbf{xA}+\mathbf{zB}=\mathbf{0},$$

$$\mathbf{z}\geqq\mathbf{0}, \tag{5.1}$$

$$\zeta_k=0\Rightarrow\xi_i=0,\quad i\in I_k,\quad k=1,\ldots,r,$$

$$\mathbf{cx}+\mathbf{dz}+F(\mathbf{x},\mathbf{z})<0. \tag{5.2}$$

Az (5.1), (5.2) rendszer a feltételek miatt nem megoldható és (5.1) *Slater-reguláris,* valamint F konvex függvény. A *konvex Farkas-tétel* alkalmazható, tehát létezik $\bar{\mathbf{y}}$ úgy, hogy tetszőleges $(\mathbf{x},\mathbf{z})\in C$ esetén

$$\mathbf{cx}+\mathbf{dz}+F(\mathbf{x},\mathbf{z})-\mathbf{xA}\bar{\mathbf{y}}-\mathbf{zB}\bar{\mathbf{y}}\geqq 0.$$

Rögzítsünk egy k indexet. Legyen $\zeta_k=1$ és $\mathbf{z}$ többi koordinátája 0, valamint $i\notin I_k$ esetén $\xi_i=0$, továbbá

$$\xi_i=\begin{cases}(\mathbf{a}_i\bar{\mathbf{y}}-\gamma_i)^{p_i-1}, & \text{ha}\quad \mathbf{a}_i\bar{\mathbf{y}}>\gamma_i,\\ -(\gamma_i-\mathbf{a}_i\bar{\mathbf{y}})^{p_i-1}, & \text{ha}\quad \mathbf{a}_i\bar{\mathbf{y}}\leqq\gamma_i,\end{cases}\qquad \text{ha}\quad i\in I_k.$$

Vezessük be a következő jelöléseket:

$$I_{k_1}=\{i\in I_k\,|\,\mathbf{a}_i\bar{\mathbf{y}}>\gamma_i\},\quad I_{k_2}=\{i\in I_k\,|\,\mathbf{a}_i\bar{\mathbf{y}}\leqq\gamma_i\}.$$

Ezt az $(\mathbf{x},\mathbf{z})$ értéket behelyettesítve a következő egyenlőtlenséget nyerjük:

$$\sum_{i\in I_{k_1}}\gamma_i(\mathbf{a}_i\bar{\mathbf{y}}-\gamma_i)^{p_i-1}+\sum_{i\in I_{k_2}}-\gamma_i(\gamma_i-\mathbf{a}_i\bar{\mathbf{y}})^{p_i-1}+\delta_k+$$

$$+\sum_{i\in I_k}\frac{1}{q_i}|\mathbf{a}_i\bar{\mathbf{y}}-\gamma_i|^{q_i(p_i-1)}-\sum_{i\in I_{k_1}}\mathbf{a}_i\bar{\mathbf{y}}(\mathbf{a}_i\bar{\mathbf{y}}-\gamma_i)^{p_i-1}-\sum_{i\in I_{k_2}}-\mathbf{a}_i\bar{\mathbf{y}}(\gamma_i-\mathbf{a}_i\bar{\mathbf{y}})^{p_i-1}-\mathbf{b}_k\bar{\mathbf{y}}\geqq 0,$$

vagy rendezve:

$$\mathbf{b}_k\bar{\mathbf{y}}-\delta_k+\sum_{i\in I_{k_1}}|\mathbf{a}_i\bar{\mathbf{y}}-\gamma_i|^{p_i}+\sum_{i\in I_{k_2}}|\gamma_i-\mathbf{a}_i\bar{\mathbf{y}}|^{p_i}-\sum_{i\in I_k}\frac{1}{q_i}|\mathbf{a}_i\bar{\mathbf{y}}-\gamma_i|^{p_i}\leqq 0,$$

amiből nyerjük, hogy

$$\sum_{i\in I_k}\frac{1}{p_i}|\mathbf{a}_i\bar{\mathbf{y}}-\gamma_i|^{p_i}+\mathbf{b}_k\bar{\mathbf{y}}-\delta_k\leqq 0,\quad \text{vagyis}\quad G_k(\bar{\mathbf{y}})\leqq 0.$$

Ezt az eljárást minden k-ra elvégezhetjük, így $\bar{\mathbf{y}}\in P$.

Véges optimumú *Slater-reguláris* duál feladat esetén a primál feladatnak van optimális megoldása. Ezt mondja a most következő, „gyenge" dualitási tétel, ami a dualitási tételek bizonyításának egyik legfontosabb segédeszköze lesz.

5.2. TÉTEL. Ha a duál feladat *Slater-reguláris* és v véges, akkor létezik $\mathbf{y}^*\in P$ úgy, hogy $\mathbf{fy}^*=v$.

Bizonyítás. Tekintsük a következő feltételrendszert:

$$\mathbf{xA}+\mathbf{zB}+\zeta_0(-\mathbf{f}) = 0,$$

$$(\zeta_0, \mathbf{z}) \geqq 0, \tag{5.3}$$

$$\zeta_k = 0 \Rightarrow \xi_i = 0, \quad i \in I_k, \quad k = 1, \ldots, r,$$

$$\mathbf{cx}+\mathbf{dz}+F(\mathbf{x}, \mathbf{z})+\zeta_0(-v) \geqq 0. \tag{5.4}$$

A tétel feltételei miatt (5.3) *Slater-reguláris.* Bebizonyítjuk, hogy (5.3) megengedett halmazán (5.4) mindig igaz. Tegyük fel indirekt, hogy $\mathbf{cx}+\mathbf{dz}+F(\mathbf{x}, \mathbf{z})+\zeta_0(-v)<0$ valamely $(\mathbf{x}, \mathbf{z}, \zeta_0)$ esetén, amely kielégíti (5.3)-at. Ekkor, ha $\zeta_0=0$, akkor

$$\mathbf{xA}+\mathbf{zB} = 0,$$

$$\mathbf{z} \geqq 0,$$

$$\zeta_k = 0 \Rightarrow \xi_i = 0, \quad i \in I_k, \quad k = 1, \ldots, r,$$

$$\mathbf{cx}+\mathbf{dz}+F(\mathbf{x}, \mathbf{z}) < 0.$$

Ekkor $(\mathbf{x}^0, \mathbf{z}^0) \in D$ esetén $(\mathbf{x}^0+\vartheta\mathbf{x}, \mathbf{z}^0+\vartheta\mathbf{z}) \in D$ a 3.5. lemma miatt, és erre a pontra $F(\mathbf{x}, \mathbf{z})$ szubadditivitása miatt

$$\mathbf{c}(\mathbf{x}^0+\vartheta\mathbf{x})+\mathbf{d}(\mathbf{z}^0+\vartheta\mathbf{z})+F(\mathbf{x}^0+\vartheta\mathbf{x}, \mathbf{z}^0+\vartheta\mathbf{z}) \leqq$$

$$\leqq \mathbf{cx}^0+\mathbf{dz}^0+F(\mathbf{x}^0, \mathbf{z}^0)+\vartheta(\mathbf{cx}+\mathbf{dz}+F(\mathbf{x}, \mathbf{z})) \to -\infty, \quad \text{ha} \quad \vartheta \to +\infty.$$

Így ellentmondásba kerültünk v végességével.

Ha $\zeta_0>0$, akkor $\left(\frac{\mathbf{x}}{\zeta_0}, \frac{\mathbf{z}}{\zeta_0}\right)$ jobb megoldás lenne az optimumnál, ami szintén lehetetlen.

Tehát (5.3) megengedett pontjaira (5.4) mindig igaz, így teljesülnek az 5.1. lemma feltételei. E szerint létezik $\mathbf{y}^*$ úgy, hogy $G_k(\mathbf{y}^*) \leqq 0$, $k=1, \ldots, r$, vagyis $\mathbf{y}^* \in P$ és $-\mathbf{fy}^*+v \leqq 0$, de $v \geqq \mathbf{fy}^*$ a fő lemma miatt, így $v=\mathbf{fy}^*$, ami azt jelenti, hogy $\mathbf{y}^*$ optimális megoldása a primál feladatnak.

KÖVETKEZMÉNY. Ha a duál feladat *Slater-reguláris,* akkor v akkor és csak akkor véges, ha $P^* \neq \varnothing$, és ekkor $\mu=v$.

Bizonyítás. Az állítás a fő lemma és az 5.2. tétel miatt nyilvánvaló.

A 4.1. tétel és az 5.2. tétel következményeit összehasonlítva láthatjuk, hogy a primál, illetve duál feladatra vonatkozóan teljes szimmetriát mutatnak. Sajnos, további vizsgálataink eredményei már nem lesznek ilyen szimmetrikusak.

6. A redukált l_p programozási feladatpár

A redukált feladatpárt abban az esetben definiáljuk, amikor $D \neq \varnothing$. Emlékeztetünk, hogy ez azt jelenti a 3.1. definícióban bevezetett Γ és $\bar{\Gamma}$ indexhalmazok esetén, hogy $0 \in \Gamma$ és ahogy ott láttuk $\Gamma_0=\Gamma-\{0\}$. Mivel a 3.5. lemma 2. következménye miatt $(\mathbf{x}, \mathbf{z}) \in D$ esetén $\zeta_k=\xi_i=0$, $i \in I_k$, $k \in \bar{\Gamma}$, így a duál feladat célfüggvény

értéke nem változik, ha abban az összegzést csak $k \in \Gamma_0$-ra írjuk elő és **A, B**, valamint **c, d**-ből is töröljük a $k \in \bar{\Gamma}$-nak megfelelő sorokat, illetve koordinátákat. Így nyerjük a redukált duál feladatot. A primál feladat esetében ez azt jelenti, hogy csak a $k \in \Gamma_0$-nak megfelelő feltételek szerepelnek.

Rendezzük át az **A, B** mátrixok sorait, a **c, d, x, z** vektorok koordinátáit és a p_i, q_i számokat úgy, hogy a $\bar{\Gamma}$-ban levő indexeknek megfelelő komponensek a Γ_0-ban levők után következzenek. Ez csak átindexelést jelent, ezt azért tegyük meg, hogy szemléletesebben fogalmazhassuk meg a redukált l_p programozási feladatpárt.

A redukált l_p programozási primál feladat:

$$G_k(\mathbf{y}) \leqq 0, \quad k \in \Gamma_0$$

$$\sup \mathbf{fy}.$$

A redukált l_p programozási duál feladat:

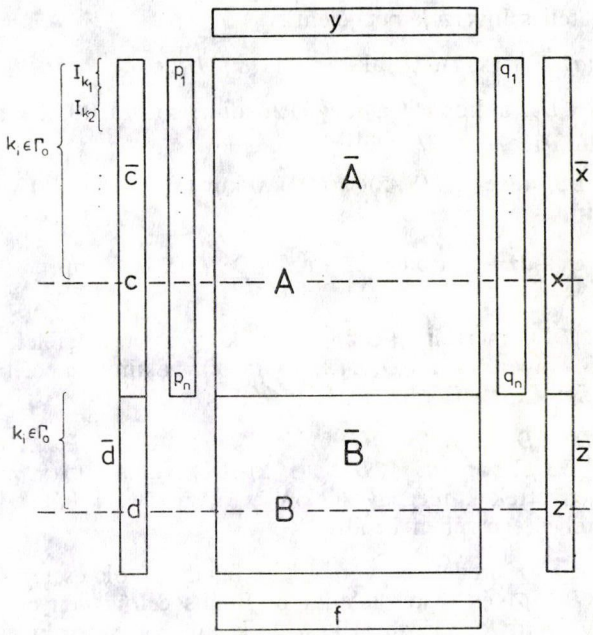
$$\bar{\mathbf{x}}\bar{\mathbf{A}} + \bar{\mathbf{z}}\bar{\mathbf{B}} = \mathbf{f}$$

$$\bar{\mathbf{z}} \geqq \mathbf{0}$$

$$\bar{\zeta}_k = 0 \Rightarrow \bar{\xi}_i = 0, \quad i \in I_k, \quad k \in \Gamma_0$$

$$\inf \left\{ \bar{\mathbf{c}}\bar{\mathbf{x}} + \bar{\mathbf{d}}\bar{\mathbf{z}} + \sum_{k \in \Gamma_0} F_k(\bar{\mathbf{x}}, \bar{\mathbf{z}}) \right\}.$$

Vezessük be a következő jelöléseket:



2. ábra

$P_r = \{\mathbf{y} | G_k(\mathbf{y}) \leqq 0,\ k \in \Gamma_0\}$,

$P_r^* = \{\mathbf{y} | \mathbf{y} \in P_r$ és optimális megoldása a redukált primál feladatnak$\}$,

$D_r = \{(\bar{\mathbf{x}}, \bar{\mathbf{z}}) | \bar{\mathbf{x}}\bar{\mathbf{A}} + \bar{\mathbf{z}}\bar{\mathbf{B}} = \mathbf{f},\ \bar{\mathbf{z}} \geqq \mathbf{0},\ \zeta_k = 0 \Rightarrow \xi_i = 0,\ i \in I_k,\ k \in \Gamma_0\}$,

$D_r^* = \{(\bar{\mathbf{x}}, \bar{\mathbf{z}}) | (\bar{\mathbf{x}}, \bar{\mathbf{z}}) \in D_r$ és optimális megoldása a redukált duál feladatnak$\}$,

$$\mu_r = \sup_{\mathbf{y} \in P_r} \mathbf{fy}, \quad v_r = \inf_{(\bar{\mathbf{x}}, \bar{\mathbf{z}}) \in D_r} \{\bar{\mathbf{c}}\bar{\mathbf{x}} + \bar{\mathbf{d}}\bar{\mathbf{z}} + \sum_{k \in \Gamma_0} F_k(\bar{\mathbf{x}}, \bar{\mathbf{z}})\}.$$

Legyen Φ az a lineáris transzformáció, melyre $\Phi(\mathbf{x}, \mathbf{z}) = (\bar{\mathbf{x}}, \bar{\mathbf{z}})$. Így Φ törli $\mathbf{x}$-ből a $k \in \bar{\Gamma}$ koordinátákat és $\mathbf{x}$-ből az $i \in I_k$, $k \in \bar{\Gamma}$ koordinátákat.

Először a redukált l_p programozási duál feladat és az eredeti duál feladat közti kapcsolatot vizsgáljuk.

Vegyük észre azt a nyilvánvaló tényt, hogy amennyiben a primál és duál feladat esetén igaz, hogy $P \neq \emptyset$ és $D \neq \emptyset$, akkor redukáltjuk esetén a redukált duál feladat *Slater-reguláris*, és a redukált duál feladat a redukált primál feladat duálisa.

6.1. LEMMA. Legyen $D \neq \emptyset$, ekkor

a) $D_r \neq \emptyset$,

b) $v = v_r$,

c) $D_r^* = \Phi(D^*)$,

d) $D^* = D \cap \Phi^{-1}(D_r^*)$,

ahol $\Phi^{-1}(D_r^*)$ a teljes inverz képet jelenti.

Bizonyítás. a) Ha $(\mathbf{x}, \mathbf{z}) \in D$, akkor $\Phi(\mathbf{x}, \mathbf{z}) \in D_r$.

b) Ha $(\bar{\mathbf{x}}, \bar{\mathbf{z}}) \in D_r$, akkor ezt a megfelelő dimenziósra 0-kal kiegészítve, az így nyert $(\mathbf{x}, \mathbf{z}) \in D$, így $v_r \succ v$, de *a)* miatt $v_r \leqq v$, tehát $v = v_r$.

c) $\Phi(D^*) \subset D_r^*$, mivel ha $(\bar{\mathbf{x}}, \bar{\mathbf{z}}) \in \Phi(D^*)$ akkor $(\bar{\mathbf{x}}, \bar{\mathbf{z}}) = \Phi(\mathbf{x}, \mathbf{z})$ valamely $(\mathbf{x}, \mathbf{z}) \in D$ esetén. Így mivel

$$v = v_r = \mathbf{cx} + \mathbf{dz} + F(\mathbf{x}, \mathbf{z}) = \bar{\mathbf{c}}\bar{\mathbf{x}} + \bar{\mathbf{d}}\bar{\mathbf{z}} + \sum_{k \in \Gamma_0} F_k(\bar{\mathbf{x}}, \bar{\mathbf{z}}), \quad \text{ezért} \quad (\bar{\mathbf{x}}, \bar{\mathbf{z}}) \in D_r^*.$$

Másrészt $D_r^* \subset \Phi(D^*)$, mert ha $(\bar{\mathbf{x}}, \bar{\mathbf{z}}) \in D_r^*$, akkor ezt a megfelelő helyeken 0-kal kiegészítve, az így nyert $(\mathbf{x}, \mathbf{z}) \in D$, $\Phi(\mathbf{x}, \mathbf{z}) = (\bar{\mathbf{x}}, \bar{\mathbf{z}})$, valamint a célfüggvény értékek is megegyeznek, így $(\mathbf{x}, \mathbf{z}) \in D^*$. Tehát $D_r^* = \Phi(D^*)$.

d) Mivel $D^* \subset D$ és $D_r^* = \Phi(D^*)$, így $D^* \subset \Phi^{-1}(D_r^*) \cap D$. Másrészt legyen $(\mathbf{x}, \mathbf{z}) \in \Phi^{-1}(D_r^*) \cap D$, ekkor $(\mathbf{x}, \mathbf{z}) \in D$, $\Phi(\mathbf{x}, \mathbf{z}) \in D_r^*$, valamint $(\mathbf{x}, \mathbf{z})$ és $\Phi(\mathbf{x}, \mathbf{z})$-hez tartozó célfüggvényértékek megegyeznek, így $(\mathbf{x}, \mathbf{z}) \in D^*$. Tehát $D^* \supset \Phi^{-1}(D_r^*) \cap D$, így lemmánkat teljes egészében beláttuk.

Belátjuk, hogy az eredeti és a redukált feladatpárok esetén $P \neq \emptyset$ akkor és csak akkor, ha $P_r \neq \emptyset$ és a két feladat optimális célfüggvényértéke megegyezik. Ezért először egy $\{\mathbf{y}^j\} \subset R^m$ sorozatot készítünk, amelynek alkalmas lineáris kombinációja segítségével nyerhetjük egy adott $\mathbf{y}^0 \in P_r$-ből P-nek egy pontját.

6.2. LEMMA. Legyen $\mathbf{b}_0 = -\mathbf{f}$. Ha $\bar{\Gamma} \neq \varnothing$, akkor létezik $\bar{\Gamma}$-nak $\Gamma_1, \ldots, \Gamma_s$ indexhalmazokra való particionálása, valamint $\mathbf{y}^j$ *Tucker-megoldása* a

$$
\begin{aligned}
&\mathbf{b}_k\mathbf{y} \geqq 0, \quad k \in \bar{\Gamma}, \\
&\mathbf{b}_k\mathbf{y} = 0, \quad k \in \Gamma, \\
&\mathbf{a}_i\mathbf{y} = 0, \quad i \in I_k, \quad k \in \Gamma \cup \left(\bar{\Gamma} - \bigcup_{v=1}^{j-1} \Gamma_v\right)
\end{aligned} \tag{6.1}
$$

egyenlőtlenségrendszernek $j=1, \ldots, s$ úgy, hogy

$$
\begin{aligned}
&\Gamma_j \neq \varnothing, \quad j = 1, \ldots, s \\
&\Gamma_i \cap \Gamma_j = \varnothing, \quad i \neq j,
\end{aligned} \tag{6.2}
$$

$$
\bar{\Gamma} = \bigcup_{i=1}^{s} \Gamma_j,
$$

továbbá

$$
\Gamma_j = \left\{k \in \bar{\Gamma} - \bigcup_{i=1}^{j-1} \Gamma_i \mid \mathbf{b}_k\mathbf{y}^j > 0\right\}, \quad j = 1, \ldots, s \tag{6.3}
$$

(A *Tucker-megoldást* a *Tucker-tételnél* definiáltuk.)

Bizonyítás. Egymás után sorban készítjük el az $\mathbf{y}^1, \Gamma_1$, majd az $\mathbf{y}^2, \Gamma_2, \ldots$ *Tucker-megoldásokat* és a hozzájuk tartozó indexhalmazokat. Mivel $\Gamma_j \neq \varnothing, j=1, \ldots, s$ igaz lesz, így eljárásunk véges számú lépésben végetér.

$j=1$. Ekkor (6.1) és a *Tucker-tételben* szereplő párja a következő formájú lesz:

$$
\begin{array}{l|l}
\mathbf{a}_i\mathbf{y} = 0, \quad i \in I_k, \quad k \in \Gamma \cup \bar{\Gamma}, & \mathbf{xA} + \mathbf{zB} = \zeta_0\mathbf{f}, \\
\mathbf{b}_k\mathbf{y} = 0, \quad k \in \Gamma, & \zeta_k \geqq 0, \quad k \in \bar{\Gamma}. \\
\mathbf{b}_k\mathbf{y} \geqq 0, \quad k \in \bar{\Gamma}. &
\end{array}
$$

Ennek a két egyenlőtlenség-rendszernek a *Tucker-tétel* miatt létezik olyan $\mathbf{y}^1, (\mathbf{x}^1, \mathbf{z}^1, \zeta_0^1)$ megoldáspárja, hogy $\mathbf{b}_k\mathbf{y}^1 + \zeta_k^1 > 0$ $k \in \bar{\Gamma}$ esetén. Ha $\Gamma_1 = \{k \in \bar{\Gamma} \mid \mathbf{b}_k\mathbf{y}^1 > 0\} = \bar{\Gamma}$, akkor kész a felbontás, $s=1$. Ha $\Gamma_1 \neq \bar{\Gamma}$, akkor tegyük fel indirekt, hogy $\Gamma_1 = \varnothing$ és így $\zeta_k^1 > 0$ $k \in \bar{\Gamma}$. Ekkor van olyan $\vartheta > 0$, hogy $(\mathbf{x}^+, \mathbf{z}^+, \zeta_0^+) + \vartheta(\mathbf{x}^1, \mathbf{z}^1, \zeta_0^1) \in D_0$ ($(\mathbf{x}^+, \mathbf{z}^+, \zeta_0^+)$-t a 3.5. lemma 2. következményében vezettük be), de ennek $\bar{\Gamma}$-ban is van pozitív koordinátája, ami ellentmond $\bar{\Gamma}$ definíciójának. Tehát $\Gamma_1 \neq \varnothing$.

$j=2$. Ekkor (6.1) és a *Tucker-tételben* szereplő párja a következő lesz:

$$
\begin{array}{l|l}
\mathbf{a}_i\mathbf{y} = 0, \quad i \in I_k, \quad k \in \Gamma \cup (\bar{\Gamma} - \Gamma_1), & \mathbf{xA} + \mathbf{zB} = \zeta_0\mathbf{f}, \\
\mathbf{b}_k\mathbf{y} = 0, \quad k \in \Gamma, & \zeta_k \geqq 0, \quad k \in \bar{\Gamma}, \\
\mathbf{b}_k\mathbf{y} \geqq 0, \quad k \in \bar{\Gamma}. & \xi_i = 0, \quad i \in I_k, \quad k \in \Gamma_1.
\end{array}
$$

Ennek a két egyenlőtlenség-rendszernek a *Tucker-tétel* miatt van olyan $\mathbf{y}^2, (\mathbf{x}^2, \mathbf{z}^2, \zeta_0^2)$ megoldáspárja, melyre $\mathbf{b}_k\mathbf{y}^2 + \zeta_k^2 > 0$, $k \in \bar{\Gamma}$. Ha $\Gamma_2 = \bar{\Gamma} - \Gamma_1$, akkor $s=2$ és kész a felbontás. Ha ez nem igaz, akkor tegyük fel indirekt, hogy $\Gamma_2 = \varnothing$. Ez azt jelenti, hogy $\zeta_k^2 > 0$, $k \in \bar{\Gamma} - \Gamma_1$. Ekkor van olyan $\vartheta > 0$, hogy $(\mathbf{x}^+, \mathbf{z}^+, \zeta_0^+) +$

$+\vartheta(\mathbf{x}^2, \mathbf{z}^2, \zeta_0^2) \in D_0$, de ennek már $\bar{\Gamma}$-ban is vannak pozitív koordinátái, ami ellentmondás. Így $\Gamma_2 \neq \emptyset$.

Az eljárást hasonlóképpen folytathatjuk, véges számú lépésben véget fog érni, mivel $\Gamma_j \neq \emptyset$ igaz minden j-re és $\bar{\Gamma}$ véges elemszámú halmaz.

Vezessük be a következő jelölést: $G_0(\mathbf{y}) = \mathbf{b}_0\mathbf{y} = -\mathbf{f}\mathbf{y}$.

6.3. TÉTEL. Legyen $\mathbf{y}^0 \in R^m$ és $L \in R$ tetszőleges, akkor létezik $\mathbf{y}^+ \in R^m$ úgy, hogy

$$G_k(\mathbf{y}^0 - \mathbf{y}^+) = G_k(\mathbf{y}^0), \quad k \in \Gamma,$$

$$G_k(\mathbf{y}^0 - \mathbf{y}^+) \leqq L, \qquad k \in \bar{\Gamma},$$

továbbá

$$\mathbf{y}^+ = \sum_{j=1}^{s} t_j \mathbf{y}^j,$$

ahol

$$t_j \geqq \max_{k \in \Gamma_j} \frac{G_k\left(\mathbf{y}^0 - \sum_{v=j+1}^{s} t_v \mathbf{y}^v\right) - L}{\mathbf{b}_k \mathbf{y}^j}, \quad j = 1, \ldots, s. \tag{6.4}$$

(Az itt szereplő $\mathbf{y}^j$, Γ_j, $j=1, \ldots, s$ a 6.2. lemmában volt definiálva.)

Bizonyítás. Igaz a következő (6.4) miatt.

$$G_k\left(\mathbf{y}^0 - \sum_{v=j+1}^{s} t_v \mathbf{y}^v\right) - t_j \mathbf{b}_k \mathbf{y}^j \leqq L, \quad k \in \Gamma_j,$$

$$\sum_{i \in I_k} \frac{1}{p_i} \left| \mathbf{a}_i \left(\mathbf{y}^0 - \sum_{v=j+1}^{s} t_v \mathbf{y}^v\right) - \gamma_i \right|^{p_i} + \mathbf{b}_k \left(\mathbf{y}^0 - \sum_{v=j}^{s} t_v \mathbf{y}^v\right) - \delta_k \leqq L, \quad k \in \Gamma_j.$$

(6.1), (6.2), (6.3) miatt

$$\mathbf{b}_k \mathbf{y}^v = 0, \quad k \in \Gamma \cup \left(\bar{\Gamma} - \bigcup_{j=1}^{v-1} \Gamma_j\right), \quad v = 1, \ldots, s,$$

$$\mathbf{a}_i \mathbf{y}^v = 0, \quad i \in I_k \quad k \in \Gamma \cup \left(\bar{\Gamma} - \bigcup_{j=1}^{v} \Gamma_j\right), \quad v = 1, \ldots, s,$$

továbbá

$$\Gamma_v \cap \Gamma_j = \emptyset, \quad v = 1, \ldots, j-1, \quad \text{így} \quad \Gamma_j \subset \bar{\Gamma} - \bigcup_{i=1}^{v} \Gamma_i, \quad v = 1, \ldots, j-1.$$

Így a zárójeleken belüli $\sum$-k kiegészíthetők, mivel az új tagok nullák. Ezt elvégezve nyerjük, hogy

$$\sum_{i \in I_k} \frac{1}{p_i} |\mathbf{a}_i(\mathbf{y}^0 - \mathbf{y}^+) - \gamma_i|^{p_i} + \mathbf{b}_k(\mathbf{y}^0 - \mathbf{y}^+) - \delta_k \leqq L, \quad k \in \Gamma_j.$$

Ez igaz $j=1, \ldots, s$ így $G_k(\mathbf{y}^0 - \mathbf{y}^+) \leqq L$, $k \in \bar{\Gamma}$, továbbá (6.1) miatt $\mathbf{a}_i \mathbf{y}^j = \mathbf{b}_k \mathbf{y}^j = 0$, $i \in I_k$, $k \in \Gamma$, $j=1, \ldots, s$, így $G_k(\mathbf{y}^0 - \mathbf{y}^+) = G_k(\mathbf{y}^0)$, $k \in \Gamma$ esetén. Így tételünket beláttuk.

KÖVETKEZMÉNY. Ha $D = \emptyset$ és $P \neq \emptyset$, akkor $\mu = +\infty$.

Bizonyítás. Legyen $D=\emptyset$, emiatt $0\in\bar{\Gamma}$ és legyen $\mathbf{y}^0\in P$. Tételünk szerint létezik $\mathbf{y}^+$ úgy, hogy $G_k(\mathbf{y}^0-\mathbf{y}^+)\leqq L$, $k\in\bar{\Gamma}$ esetén és $G_k(\mathbf{y}^0-\mathbf{y}^+)=G_k(\mathbf{y}^0)\leqq 0$, $k\in\Gamma$. Ha tetszőleges $L\leqq 0$ számot választunk, akkor $G_k(\mathbf{y}^0-\mathbf{y}^+)\leqq 0$, $k=1, \ldots, r$ és $f(\mathbf{y}^0-\mathbf{y}^+)\geqq -L$. Így $\mu=\sup_{\mathbf{y}\in P}\mathbf{fy}=+\infty$.

A primál és a redukált primál feladat kapcsolatának jellemzésére rendelkezésünkre áll a legfontosabb segédeszköz, amit 6.3. tétel biztosít.

6.4. LEMMA. Legyen $D\neq\emptyset$, ekkor

a) $P\neq\emptyset$ akkor és csak akkor, ha $P_r\neq\emptyset$,

b) $\mu=\mu_r$,

c) $P^*\subset P_r^*$ és minden $\mathbf{y}\in P_r^*$-hoz létezik $\mathbf{y}^+$ úgy, hogy $\mathbf{y}-\mathbf{y}^+\in P^*$.

Bizonyítás. a) Ha $\mathbf{y}\in P$, akkor $\mathbf{y}\in P_r$ mivel csak feltételeket hagytunk el. Ha $\mathbf{y}\in P_r$, akkor $G_k(\mathbf{y})\leqq 0$, $k\in\Gamma_0$ esetén és a 6.3. tétel miatt létezik $\mathbf{y}^+$ úgy, hogy $G_k(\mathbf{y}-\mathbf{y}^+)\leqq 0$, $k\in\Gamma_0\cup\bar{\Gamma}$ esetén, $L=0$ választás miatt. Továbbá ekkor $\mathbf{fy}^+=0$, mivel $D\neq\emptyset$.

b) Nyilvánvaló *a)* bizonyítása miatt, hogy $\mu=\mu_r$.

c) Ha $\mathbf{y}^*\in P^*$, akkor $\mathbf{y}^*\in P_r$ és mivel $\mu=\mu_r$ így $\mathbf{y}^*\in P_r^*$. Ha $\mathbf{y}^*\in P_r^*$ akkor a 6.3. tétel miatt $L=0$-hoz létezik $\mathbf{y}^+$ úgy, hogy $G_k(\mathbf{y}^*-\mathbf{y}^+)\leqq 0$, $k=1, \ldots, r$ és $\mathbf{f}(\mathbf{y}^*-\mathbf{y}^+)=\mathbf{fy}^*$, így $\mathbf{y}^*-\mathbf{y}^+\in P^*$.

6.5. LEMMA. A primál feladat akkor és csak akkor *Slater-reguláris,* amikor redukáltja is az.

Bizonyítás. Ha a primál feladat *Slater-reguláris,* akkor nyilván redukáltja is az. Ha a redukált primál feladat *Slater-reguláris,* akkor $L=-1$ mellett a redukált feladat egy *Slater-pontjára* alkalmazva a 6.3. tételt, azt nyerjük, hogy az eredeti feladat is *Slater-reguláris.*

7. Dualitás tételek

Ebben a fejezetben először a célfüggvények korlátosságának feltételeit vizsgáljuk, majd az l_p programozás „erős" dualitás tételét bizonyítjuk. Két lemmában pedig az optimális megoldások halmazait jellemezzük.

7.1. TÉTEL. *a)* Tegyük fel, hogy $P\neq\emptyset$. Ebben az esetben μ akkor és csak akkor véges, ha $D\neq\emptyset$.

b) Tegyük fel, hogy $D\neq\emptyset$. Ebben az esetben v akkor és csak akkor véges, ha $P\neq\emptyset$.

Bizonyítás. a) Ha P sem és D sem üres, akkor a fő lemma miatt μ véges. Ha μ véges, akkor a 6.3. tétel következménye miatt $D=\emptyset$ nem lehetséges, mert akkor $\mu=+\infty$. Így tehát $D\neq\emptyset$.

b) Ha P sem és D sem üres, akkor a fő lemma miatt v véges. Ha v véges, akkor redukáljuk a feladatpárt, ekkor a redukált duál feladat *Slater-reguláris,* $v=v_r$, így az 5.3. tétel miatt P_r nem üres, továbbá a 6.4. lemma miatt P sem üres.

7.2. TÉTEL. Tegyük fel, hogy $P\neq\emptyset$ és $D\neq\emptyset$, akkor $\mu=v$.

Bizonyítás. A fő lemma miatt μ is és v is véges. Redukáljuk a feladatpárt. Ekkor a 6.1. lemma miatt $v=v_r$ és a 6.4. lemma miatt $\mu=\mu_r$. A redukált duál feladat *Slater-reguláris*, így az 5.2. tétel miatt $\mu_r=v_r$ és így $\mu=v$.

7.3. TÉTEL. Tegyük fel, hogy $P\neq\varnothing$ és μ véges, akkor $P^*\neq\varnothing$.

Bizonyítás. A 7.1. és 7.2. tétel miatt ekkor $D\neq\varnothing$, valamint $\mu=v=v_r=\mu_r$, továbbá az 5.2. tétel miatt $P_r^*\neq\varnothing$ és ekkor a 6.4. lemma miatt az l_p programozási primál feladatnak is van optimális megoldása.

Azt jelenti a 7.3. tétel, hogy a primál feladat megfogalmazásában suprémum helyett maximum írható, mivel ha $P\neq\varnothing$ és μ véges, akkor a primál feladat mindig felveszi az optimális értékét.

1. KÖVETKEZMÉNY. Legyen $\mathbf{y}^*\in P^*$ és $(\mathbf{x}, \mathbf{z})\in D$, ekkor $(\mathbf{x}, \mathbf{z})\in D^*$ akkor és csak akkor igaz, ha

$$\zeta_k G_k(\mathbf{y}^*)=0, \quad k=1, \ldots, r,$$

$$\zeta_k \operatorname{sgn}(\mathbf{a}_i\mathbf{y}^*-\gamma_i)|\mathbf{a}_i\mathbf{y}^*-\gamma_i|^{p_i-1}=\xi_i, \quad i\in I_k, \quad k=1, \ldots, r,$$

vagy ami ezzel ekvivalens

$$\zeta_k G_k(\mathbf{y}^*)=0, \quad k=1, \ldots, r,$$

$$\zeta_k=0 \quad \text{vagy} \quad \mathbf{a}_i\mathbf{y}^*-\gamma_i=\operatorname{sgn}\xi_i\left|\frac{\xi_i}{\zeta_k}\right|^{q_i-1}, \quad i\in I_k, \quad k=1, \ldots, r,$$

továbbá D^* zárt félterek metszete.

Bizonyítás. A fő lemma és a 7.2. tétel miatt $(\mathbf{x}, \mathbf{z})$ akkor és csak akkor optimális, ha a feltételek igazak. D^* zárt félterek metszeteként írható fel, ugyanis legyen $\Delta(\mathbf{y}^*)=\{k\in\{1, \ldots, r\}|G_k(\mathbf{y}^*)<0\}$, $\bar{\Delta}(\mathbf{y}^*)=\{1, \ldots, r\}-\Delta(\mathbf{y}^*)=\{k\in\{1, \ldots, r\}|G_k(\mathbf{y}^*)=0\}$ és ekkor D^* a következő félterek metszeteként írható fel:

$$D^*=\left\{(\mathbf{x}, \mathbf{z}) \;\middle|\; \begin{array}{l} \mathbf{x}\mathbf{A}+\mathbf{z}\mathbf{B}=\mathbf{f}, \\ \zeta_k=0, \quad k\in\Delta(\mathbf{y}^*), \\ \xi_i=0, \quad i\in I_k, \quad k\in\Delta(\mathbf{y}^*), \\ \zeta_k=0, \quad k\in\bar{\Delta}(\mathbf{y}^*), \\ \xi_i=\zeta_k\operatorname{sgn}(\mathbf{a}_i\mathbf{y}^*-\gamma_i)|\mathbf{a}_i\mathbf{y}^*-\gamma_i|^{p_i-1}, \quad i\in I_k, \quad k\in\bar{\Delta}(\mathbf{y}^*). \end{array}\right\}.$$

2. KÖVETKEZMÉNY. Legyen $(\mathbf{x}^*, \mathbf{z}^*)\in D^*$ és $\mathbf{y}\in P$, ekkor $\mathbf{y}\in P^*$, akkor és csak akkor, ha

$$\zeta_k^* G_k(\mathbf{y})=0, \quad k=1, \ldots, r$$

$$\zeta_k^* \operatorname{sgn}(\mathbf{a}_i\mathbf{y}-\gamma_i)|\mathbf{a}_i\mathbf{y}-\gamma_i|^{p_i-1}=\xi_i^*, \quad i\in I_k, \quad k=1, \ldots, r,$$

vagy ami ezzel ekvivalens:

$$\zeta_k^* G_k(\mathbf{y})=0, \quad k=1, \ldots, r$$

$$\zeta_k^*=0, \quad \text{vagy} \quad \mathbf{a}_i\mathbf{y}-\gamma_i=\operatorname{sgn}\xi_i^*\left|\frac{\xi_i^*}{\zeta_k^*}\right|^{q_i-1}, \quad i\in I_k, \quad k=1, \ldots, r.$$

Bizonyítás. A fő lemma és a 7.3. tétel miatt igaz.

Megjegyzés. Az sajnos nem igaz, hogy mindig léteznek olyan $\mathbf{y}^*, (\mathbf{x}^*, \mathbf{z}^*)$ optimális megoldások, hogy $\zeta_k^* - G_k(\mathbf{y}^*) > 0$, $k = 1, \ldots, r$. Például tekintsük a következő feladatpárt:

$$\begin{array}{l|l} \frac{1}{2}\eta_1^2 + \frac{1}{2}\eta_2^2 + \eta_1 + \eta_2 - 1{,}25 \leqq 0 & \xi_1 + \quad \zeta_1 + \zeta_2 = 1 \\ \eta_1 - \eta_2 \leqq 0 & \xi_2 + \zeta_1 - \zeta_2 = 1 \\ \max(\eta_1 + \eta_2) & \zeta_1 \geqq 0, \quad \zeta_2 \geqq 0 \\ & \zeta_1 = 0 \Rightarrow \xi_1 = \xi_2 = 0 \\ & \min\left\{1{,}25\zeta_1 + \frac{1}{2\zeta_1}(\xi_1^2 + \xi_2^2)\right\} \end{array}$$

Könnyen belátható, hogy a primál feladatnak csak az $\mathbf{y}^* = (0{,}5;\ 0{,}5)$, a duál feladatnak csak az $(\mathbf{x}^*, \mathbf{z}^*) = \left(\frac{1}{3}, \frac{1}{3}, \frac{2}{3}, 0\right)$ az egyetlen optimális megoldása, és ekkor $\zeta_2^* - G_2(\mathbf{y}^*) = 0$.

Ebben a fejezetben bizonyított tételeinkkel bemutattuk, hogy erős dualitási kapcsolat áll fenn az l_p programozási primál és duál feladat között. Megmutattuk, hogy optimális értékeik megegyeznek amennyiben mindkét feladatnak van lehetséges megoldása, sőt azt is beláttuk, hogy ekkor a primál feladatnak mindig van optimális megoldása. A hátralevő pontok más irányú vizsgálatokat tartalmaznak, az optimális megoldások és a *Lagrange-függvények* nyeregpontjai közötti kapcsolatot mutatják be, vizsgáljuk az l_p programozási feladatok regularitását, majd érzékenységét.

8. Az l_p programozási feladatok Lagrange-függvényei

Először az l_p programozási primál feladat *Lagrange-függvényét* definiáljuk, majd az optimális megoldások és a *Lagrange-függvény* nyeregpontja közötti kapcsolatot mutatjuk be. Utána hasonlóan járunk el a duál feladat esetében.

8.1. DEFINÍCIÓ. Az l_p programozási primál feladat *Lagrange-függvényének* az $L(\mathbf{y}, \mathbf{z}) = -\mathbf{f}\mathbf{y} + \sum_{k=1}^{r} \zeta_k G_k(\mathbf{y})$ függvényt nevezzük, ahol $\mathbf{y} \in R^m$ és $\mathbf{z} = (\zeta_1, \ldots, \zeta_r) \geqq \mathbf{0}$ tetszőleges.

8.2. DEFINÍCIÓ. Az $(\bar{\mathbf{y}}, \bar{\mathbf{z}})$ pont $\bar{\mathbf{z}} \geqq 0$ nyeregpontja $L(\mathbf{y}, \mathbf{z})$-nek, ha tetszőleges $\mathbf{y} \in R^m$, $\mathbf{z} \in R^r$, $\mathbf{z} \geqq 0$ esetén

$$L(\bar{\mathbf{y}}, \mathbf{z}) \leqq L(\bar{\mathbf{y}}, \bar{\mathbf{z}}) \leqq L(\mathbf{y}, \bar{\mathbf{z}}).$$

8.1. TÉTEL. Az l_p programozási feladatpárnak $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$ és $\bar{\mathbf{y}}$ akkor és csak akkor optimális megoldásai, ha $(\bar{\mathbf{y}}, \bar{\mathbf{z}})$ nyeregpontja $L(\mathbf{y}, \mathbf{z})$-nek és

$$\bar{\xi}_i = \bar{\zeta}_k \operatorname{sgn}(\mathbf{a}_i\bar{\mathbf{y}} - \gamma_i)|\mathbf{a}_i\bar{\mathbf{y}} - \gamma_i|^{p_i - 1}, \quad i \in I_k, \quad k = 1, \ldots, r.$$

Bizonyítás. Ha $(\bar{\mathbf{y}}, \bar{\mathbf{z}})$ nyeregpontja L-nek, akkor $\bar{\mathbf{y}}$ nyilván optimális megoldása a primál feladatnak, továbbá $\mathbf{f}\bar{\mathbf{y}} = L(\bar{\mathbf{y}}, \bar{\mathbf{z}})$ miatt $\sum_{k=1}^{r} \bar{\zeta}_k G_k(\bar{\mathbf{y}}) = 0$ és így $\bar{\zeta}_k G_k(\bar{\mathbf{y}}) = 0$,

mivel $\bar{\zeta}_k \geqq 0$ és $G_k(\bar{\mathbf{y}}) \leqq 0$, $k=1, \ldots, r$. Mivel $(\bar{\mathbf{y}}, \bar{\mathbf{z}})$ nyeregpont, így $\nabla_y L(\bar{\mathbf{y}}, \bar{\mathbf{z}}) = \mathbf{0}$, vagyis

$$\mathbf{f} = \sum_{k=1}^{r} \bar{\zeta}_k \Big[\sum_{i \in I_k} \operatorname{sgn}(\mathbf{a}_i \bar{\mathbf{y}} - \gamma_i) |\mathbf{a}_i \bar{\mathbf{y}} - \gamma_i|^{p_i - 1} \mathbf{a}_i + \mathbf{b}_k \Big].$$

Legyen $\bar{\xi}_i = \operatorname{sgn}(\mathbf{a}_i \bar{\mathbf{y}} - \gamma_i) |\mathbf{a}_i \bar{\mathbf{y}} - \gamma_i|^{p_i - 1} \bar{\zeta}_k$, $i \in I_k$, $k=1, \ldots, r$, ekkor nyilván $(\bar{\mathbf{x}}, \bar{\mathbf{z}}) \in D$ és a 7.3. tétel következménye miatt $(\bar{\mathbf{x}}, \bar{\mathbf{z}}) \in D^*$ is igaz.

Fordítva, ha $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$ és $\bar{\mathbf{y}}$ optimális megoldása az l_p programozási duál, illetve primál feladatnak, akkor a fő lemma és a 7.2. tétel miatt $\bar{\zeta}_k G_k(\bar{\mathbf{y}}) = \mathbf{0}$ és

$$\bar{\xi}_i = \bar{\zeta}_k \operatorname{sgn}(\mathbf{a}_i \bar{\mathbf{y}} - \gamma_i) |\mathbf{a}_i \bar{\mathbf{y}} - \gamma_i|^{p_i - 1}, \quad i \in I_k, \quad k = 1, \ldots, r.$$

Mivel $\bar{\mathbf{x}}\mathbf{A} + \bar{\mathbf{z}}\mathbf{B} = \mathbf{f}$, így $\nabla_y L(\bar{\mathbf{y}}, \bar{\mathbf{z}}) = \mathbf{0}$. Továbbá $L(\mathbf{y}, \mathbf{z})$ konvex függvénye $\mathbf{y}$-nak, így igaz, hogy

$$L(\bar{\mathbf{y}}, \bar{\mathbf{z}}) + \nabla_y L(\bar{\mathbf{y}}, \bar{\mathbf{z}})(\mathbf{y} - \bar{\mathbf{y}}) \leqq L(\mathbf{y}, \bar{\mathbf{z}}).$$

Mivel $\nabla_y L(\bar{\mathbf{y}}, \bar{\mathbf{z}}) = \mathbf{0}$, így $L(\bar{\mathbf{y}}, \bar{\mathbf{z}}) \leqq L(\mathbf{y}, \bar{\mathbf{z}})$ és $L(\bar{\mathbf{y}}, \mathbf{z}) \geqq L(\bar{\mathbf{y}}, \bar{\mathbf{z}})$ minden $\mathbf{z} \geqq \mathbf{0}$ esetén $G_k(\bar{\mathbf{y}}) \leqq 0$ miatt, így tehát $(\bar{\mathbf{y}}, \bar{\mathbf{z}})$ nyeregpontja L-nek.

8.3. DEFINÍCIÓ. Az l_p programozási duál feladat *Lagrange-függvényének* a

$$\Psi(\mathbf{x}, \mathbf{z}, \mathbf{y}) = \mathbf{cx} + \mathbf{dz} + F(\mathbf{x}, \mathbf{z}) + \mathbf{fy} - (\mathbf{xA} + \mathbf{zB})\mathbf{y}$$

függvényt nevezzük, amely a

$$C = \{(\mathbf{x}, \mathbf{z}, \mathbf{y}) | \zeta_k \geqq 0, \ \zeta_k = 0 \Rightarrow \xi_i = 0, \quad i \in I_k, \ k = 1, \ldots, r\}$$

konvex halmazon van értelmezve.

8.4. DEFINÍCIÓ. Az $(\bar{\mathbf{x}}, \bar{\mathbf{z}}, \bar{\mathbf{y}})$ pont nyeregpontja Ψ-nek, ha tetszőleges $(\mathbf{x}, \mathbf{z}, \mathbf{y}) \in C$ esetén

$$\Psi(\bar{\mathbf{x}}, \bar{\mathbf{z}}, \mathbf{y}) \leqq \Psi(\bar{\mathbf{x}}, \bar{\mathbf{z}}, \bar{\mathbf{y}}) \leqq \Psi(\mathbf{x}, \mathbf{z}, \bar{\mathbf{y}}).$$

8.2. TÉTEL. Az l_p programozási feladatpárnak $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$ és $\bar{\mathbf{y}}$ akkor és csak akkor optimális megoldásai, ha $(\bar{\mathbf{x}}, \bar{\mathbf{z}}, \bar{\mathbf{y}})$ nyeregpontja Ψ-nek.

Bizonyítás. Ha optimális megoldások, akkor $G_k(\bar{\mathbf{y}}) \leqq 0$, vagyis

$$\sum_{i \in I_k} \frac{1}{p_i} |\mathbf{a}_i \bar{\mathbf{y}} - \gamma_i|^{p_i} + \mathbf{b}_k \bar{\mathbf{y}} - \delta_k \leqq 0, \quad k = 1, \ldots, r.$$

Legyen $(\mathbf{x}, \mathbf{z}, \bar{\mathbf{y}}) \in C$ tetszőleges. Alkalmazzuk a speciális geometriai egyenlőtlenséget $\alpha = \mathbf{a}_i \bar{\mathbf{y}} - \gamma_i$, $\beta = \frac{\xi_i}{\zeta_k}$ helyettesítéssel, ahol $i \in I_k$, és k olyan, hogy $\zeta_k > 0$. Így nyerjük, hogy

$$\sum_{i \in I_k} (\mathbf{a}_i \bar{\mathbf{y}} - \gamma_i) \frac{\xi_i}{\zeta_k} - \sum_{i \in I_k} \frac{1}{q_i} \left| \frac{\xi_i}{\zeta_k} \right|^{q_i} + \mathbf{b}_k \bar{\mathbf{y}} - \delta_k \leqq 0, \quad \text{ha} \quad k\text{-ra } \zeta_k > 0.$$

Ezeket az egyenlőtlenségeket szorozzuk meg $\zeta_k > 0$-val, összegezzük őket és egészítsük ki $(\mathbf{x}, \mathbf{z})$-t a megfelelő helyeken 0 koordinátákkal, így a következő egyenlőtlenséget nyerjük:

$$\mathbf{x}\mathbf{A}\bar{\mathbf{y}} - \mathbf{cx} - F(\mathbf{x}, \mathbf{z}) + \mathbf{z}\mathbf{B}\bar{\mathbf{y}} - \mathbf{dz} \leqq 0.$$

Ezt átalakítva

$$0 \leqq \mathbf{cx} + \mathbf{dz} + F(\mathbf{x}, \mathbf{z}) - (\mathbf{xA} + \mathbf{zB})\bar{\mathbf{y}}. \tag{8.1}$$

Továbbá tudjuk, hogy

$$\mathbf{c}\bar{\mathbf{x}}+\mathbf{d}\bar{\mathbf{z}}+F(\bar{\mathbf{x}},\bar{\mathbf{z}})=\mathbf{f}\bar{\mathbf{y}}, \tag{8.2}$$

$$\mathbf{f}\mathbf{y}-(\bar{\mathbf{x}}\mathbf{A}+\bar{\mathbf{z}}\mathbf{B})\mathbf{y}=0. \tag{8.3}$$

Adjuk össze (8.1), (8.2), (8.3)-at:

$$\mathbf{c}\bar{\mathbf{x}}+\mathbf{d}\bar{\mathbf{z}}+F(\bar{\mathbf{x}},\bar{\mathbf{z}})+\mathbf{f}\mathbf{y}-(\bar{\mathbf{x}}\mathbf{A}+\bar{\mathbf{z}}\mathbf{B})\mathbf{y}\leqq\mathbf{c}\mathbf{x}+\mathbf{d}\mathbf{z}+F(\mathbf{x},\mathbf{z})-(\mathbf{x}\mathbf{A}+\mathbf{z}\mathbf{B})\bar{\mathbf{y}}+\mathbf{f}\bar{\mathbf{y}}.$$

Így $(\bar{\mathbf{x}}, \bar{\mathbf{z}}, \bar{\mathbf{y}})$ nyeregpontja Ψ-nek.

Fordítva, ha $(\bar{\mathbf{x}}, \bar{\mathbf{z}}, \bar{\mathbf{y}})$ nyeregpontja Ψ-nek, akkor a nyeregpont-egyenlőtlenséget $(\bar{\mathbf{x}}, \bar{\mathbf{z}}, \mathbf{y})\in C$-re felírva

$$(\mathbf{f}-\bar{\mathbf{x}}\mathbf{A}-\bar{\mathbf{z}}\mathbf{B})\mathbf{y}\leqq(\mathbf{f}-\bar{\mathbf{x}}\mathbf{A}-\bar{\mathbf{z}}\mathbf{B})\bar{\mathbf{y}}.$$

Mivel itt a jobb oldalon egy fix érték áll, így $\mathbf{f}=\bar{\mathbf{x}}\mathbf{A}+\bar{\mathbf{z}}\mathbf{B}$, mert különben található lenne olyan $\mathbf{y}$, hogy nem állna fenn a fenti egyenlőtlenség. Tehát $(\bar{\mathbf{x}}, \bar{\mathbf{z}})\in D$. Legyen $(\mathbf{x}, \mathbf{z}, \bar{\mathbf{y}})\in C$, ekkor $(\bar{\mathbf{x}}+\mathbf{x}, \bar{\mathbf{z}}+\mathbf{z}, \bar{\mathbf{y}})\in C$. Erre írjuk most fel a nyeregpont-egyenlőtlenséget:

$$\mathbf{c}\bar{\mathbf{x}}+\mathbf{d}\bar{\mathbf{z}}+F(\bar{\mathbf{x}},\bar{\mathbf{z}})+\mathbf{f}\bar{\mathbf{y}}-(\bar{\mathbf{x}}\mathbf{A}+\bar{\mathbf{z}}\mathbf{B})\bar{\mathbf{y}}\leqq\mathbf{c}\bar{\mathbf{x}}+\mathbf{c}\mathbf{x}+\mathbf{d}\bar{\mathbf{z}}+\mathbf{d}\mathbf{z}+F(\bar{\mathbf{x}}+\mathbf{x},\bar{\mathbf{z}}+\mathbf{z})+\mathbf{f}\bar{\mathbf{y}}-$$

$$-[(\bar{\mathbf{x}}+\mathbf{x})\mathbf{A}+(\bar{\mathbf{z}}+\mathbf{z})\mathbf{B}]\mathbf{y}.$$

F szubadditív, így $0\leqq\mathbf{c}\mathbf{x}+\mathbf{d}\mathbf{z}+F(\mathbf{x},\mathbf{z})-(\mathbf{x}\mathbf{A}+\mathbf{z}\mathbf{B})\bar{\mathbf{y}}$ és ebből az 5.1. lemmában bemutatott eljárással azonos módon nyerjük, hogy $G_k(\bar{\mathbf{y}})\leqq 0$, $k=1, \ldots, r$, így $\bar{\mathbf{y}}\in P$.

Végül a nyeregpont-egyenlőtlenséget $\mathbf{y}=\mathbf{0}$, $\mathbf{x}=\mathbf{0}$, $\mathbf{z}=\mathbf{0}$ mellett felírva nyerjük, hogy $\mathbf{c}\bar{\mathbf{x}}+\mathbf{d}\bar{\mathbf{z}}+F(\bar{\mathbf{x}},\bar{\mathbf{z}})\leqq\mathbf{f}\bar{\mathbf{y}}$, de a fő lemma miatt a fordított egyenlőtlenség mindig fennáll, így a két célfüggvény érték megegyezik. Tehát $\bar{\mathbf{y}}$, $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$ optimális megoldáspár.

9. Regularitás az l_p programozásban

A regularitás segítségével a primál és a duál feladat optimális megoldásainak halmazát jellemezzük. A regularitás egyúttal az l_p programozási feladatpár érzékenységvizsgálatában is fontos szerepet játszik.

9.1. Definíció. Az l_p programozási primál feladatot regulárisnak nevezzük, ha az

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

$$\mathbf{B}\mathbf{y}\leqq\mathbf{0}\quad \mathbf{y}\neq\mathbf{0}$$

$$\mathbf{f}\mathbf{y}\geqq 0$$

rendszernek nincs megoldása.

9.2. Definíció. Az l_p programozási duál feladatot regulárisnak nevezzük, ha az

$$\mathbf{x}\mathbf{A}+\mathbf{z}\mathbf{B}=\mathbf{0}$$

$$\mathbf{z}\geqq\mathbf{0}\quad \mathbf{z}\neq\mathbf{0}$$

$$\zeta_k=0\Rightarrow\xi_i=0,\quad i\in I_k,\quad k=1, \ldots, r$$

$$\mathbf{c}\mathbf{x}+\mathbf{d}\mathbf{z}+F(\mathbf{x},\mathbf{z})\leqq 0$$

rendszernek nincs megoldása.

9.3. DEFINÍCIÓ. Az l_p programozási feladatpár reguláris, ha a primál és a duál feladat is reguláris.

9.4. DEFINÍCIÓ. Az l_p programozás primál feladata szuperkonzisztens, ha van olyan $\mathbf{y} \in R^m$, hogy

$$G_k(\mathbf{y}) < 0, \quad k = 1, \ldots, r.$$

Megjegyzés: Ez a feltétel erősebb a *Slater-regularitási feltételnél,* mivel ott szigorú egyenlőtlenséget csak a nemlineáris G_k függvényekre követeltünk meg.

9.1. TÉTEL. *a)* Ha az l_p programozás primál feladata reguláris, akkor a duál feladat *Slater-reguláris.*

b) Az l_p programozás duál feladata akkor és csak akkor reguláris, ha a primál feladat szuperkonzisztens.

Bizonyítás. a) Ha az l_p programozás primál feladata reguláris, akkor a 9.1. definíció miatt a

$$-\mathbf{Ay} \leqq \mathbf{0}$$

$$\mathbf{Ay} \leqq \mathbf{0}$$

$$\mathbf{By} \leqq \mathbf{0}$$

$$-\mathbf{fy} \leqq \mathbf{0}$$

rendszernek csak a triviális $\mathbf{y}=\mathbf{0}$ megoldása létezik. Ekkor a $(-\mathbf{a}_1, \ldots, -\mathbf{a}_n, \mathbf{a}_1, \ldots, \mathbf{a}_n, \mathbf{b}_1, \ldots, \mathbf{b}_r, -\mathbf{f})$ vektorok által kifeszített kúp az egész tér. Így létezik $\xi_i', \xi_i'' \geqq 0$, $i=1, \ldots, r$, $\zeta_k' \geqq 0$, $k=1, \ldots, r$ és $\vartheta \geqq 0$ számok úgy, hogy

$$\mathbf{f} - (\mathbf{b}_1 + \ldots + \mathbf{b}_r) = \sum_{i=1}^{n} \mathbf{a}_i(\xi_i' - \xi_i'') + \sum_{k=1}^{r} \mathbf{b}_k \zeta_k' + \vartheta(-\mathbf{f}).$$

Kifejezve f-et

$$\mathbf{f} = \sum_{i=1}^{n} \mathbf{a}_i \frac{\xi_i' - \xi_i''}{1+\vartheta} + \sum_{k=1}^{r} \mathbf{b}_k \frac{1+\zeta_k'}{1+\vartheta}.$$

Legyen $\xi_i = \frac{\xi_i' - \xi_i''}{1+\vartheta}$, $i=1, \ldots, n$, $\zeta_k = \frac{1+\zeta_k'}{1+\vartheta}$, $k=1, \ldots, r$, ekkor $(\mathbf{x}, \mathbf{z}) \in D$, $\mathbf{z} > \mathbf{0}$ vagyis a duál feladat kielégíti a *Slater-regularitási feltételt.*

b) Ha az l_p programozás duál feladata reguláris, akkor az

$$\mathbf{xA} + \mathbf{zB} = \mathbf{0}$$

$$\mathbf{z} \geqq \mathbf{0}$$

$$\sum_{k=1}^{r} \zeta_k = 1 \tag{9.1}$$

$$\zeta_k = 0 \Rightarrow \xi_i = 0, \quad i \in I_k, \quad k = 1, \ldots, r$$

$$\inf \{\mathbf{cx} + \mathbf{dz} + F(\mathbf{x}, \mathbf{z})\} \tag{9.2}$$

feladat esetén vagy üres a feltételi halmaz, vagy minden, a feltételeket kielégítő $(\mathbf{x}, \mathbf{z})$ esetén

$$\mathbf{cx}+\mathbf{dz}+F(\mathbf{x}, \mathbf{z}) > 0.$$

Írjuk fel a feladat primál párját.

$$\sum_{i\in I_k} \frac{1}{p_i} |(\mathbf{a}_i, 0)(\mathbf{y}, \vartheta)-\gamma_i|^{p_i}+(\mathbf{b}_k, 1)(\mathbf{y}, \vartheta)-\delta_k \leqq 0, \quad k = 1, \ldots, r$$

$$\max (0, 1)(\mathbf{y}, \vartheta)$$

Ugyanez átalakítva:

$$\sum_{i\in I_k} \frac{1}{p_i} |\mathbf{a}_i\mathbf{y}-\gamma_i|^{p_i}+\mathbf{b}_k\mathbf{y}-\delta_k+\vartheta \leqq 0, \quad k = 1, \ldots, r \tag{9.3}$$

$$\max \vartheta.$$

Ha a duál feltételi halmaz üres, akkor mivel (9.3)-ra mindig igaz, hogy $P\neq\varnothing$, így a 6.3. tétel következménye miatt $\sup \vartheta = +\infty$, tehát van olyan $(\mathbf{y}, \vartheta_0)\in P$ úgy, hogy $\vartheta_0>0$.

Ha a duál feltételi halmaz nem üres, akkor legyen $\max \vartheta=\vartheta_0$. Belátjuk, hogy $\vartheta_0>0$. Mivel tetszőleges, a (9.1) feltételeket kielégítő $(\mathbf{x}, \mathbf{z})$ esetén $\mathbf{cx}+\mathbf{dz}+F(\mathbf{x}, \mathbf{z})>0$ a duál feladat regularitása miatt, így $v=\inf \{\mathbf{cx}+\mathbf{dz}+F(\mathbf{x}, \mathbf{z})\}\geqq 0$. Tudjuk a 7.2. tételből, hogy $v=\vartheta_0$. Tegyük fel indirekt, hogy $0=\vartheta_0=v$. Ekkor létezik $(\mathbf{x}^t, \mathbf{z}^t)$, $t=1, 2, \ldots$, a (9.1) feltételeket kielégítő sorozat úgy, hogy $\mathbf{cx}^t+\mathbf{dz}^t+F(\mathbf{x}^t, \mathbf{z}^t)\to v=0$ $t\to+\infty$ esetén. (9.1)-ből következik, hogy $\|\mathbf{z}^t\|=1$, így a $\mathbf{z}^t$ sorozat korlátos. Az $\mathbf{x}^t$ sorozat is korlátos, mert $\|\mathbf{x}^t\|\to+\infty$, $\|\mathbf{z}^t\|=1$, $t\to+\infty$-ből következik, hogy $\mathbf{cx}^t+\mathbf{dz}^t+F(\mathbf{x}^t, \mathbf{z}^t)\to+\infty$, ami ellentmond $(\mathbf{x}^t, \mathbf{z}^t)$ megválasztásának.

Így az $(\mathbf{x}^t, \mathbf{z}^t)$ sorozat korlátos, kiválasztható belőle konvergens részsorozat, melynek limeszpontját jelöljük $(\mathbf{x}, \mathbf{z})$-vel. Erre igaz az, hogy $\zeta_k=0\Rightarrow\xi_i=0$, $i\in I_k$, $k=1, \ldots, r$, mivel ha nem lenne igaz, akkor valamely k-ra és $i\in I_k$-ra $\zeta_k^t\to 0$, de $\xi_i^t\to 0$. Ebből pedig közvetlenül következik, hogy $\mathbf{cx}^t+\mathbf{dz}^t+F(\mathbf{x}^t, \mathbf{z}^t)\to+\infty$, ami ellentmondás. Tehát az $(\mathbf{x}, \mathbf{z})$ limeszpontra fennállnak a következők:

$$\mathbf{xA}+\mathbf{zB} = \mathbf{0}$$

$$\mathbf{z} \geqq \mathbf{0}, \quad \|\mathbf{z}\| = 1$$

$$\zeta_k = 0 \Rightarrow \xi_i = 0, \quad i\in I_k, \quad k = 1, \ldots, r$$

$$\mathbf{cx}+\mathbf{dz}+F(\mathbf{x}, \mathbf{z}) = 0.$$

Ez ellentmond a duál feladat regularitásának, így $v=\vartheta_0>0$, tehát a 7.3. tétel miatt van olyan, a (9.3) feladat feltételeit kielégítő $(\mathbf{y}, \vartheta_0)$, hogy $\vartheta_0>0$. Erre a (9.3) feladat feltételeit felírva:

$$\sum_{i\in I_k} \frac{1}{p_i} |\mathbf{a}_i\mathbf{y}-\gamma_i|^{p_i}+\mathbf{b}_k\mathbf{y}-\delta_k+\vartheta_0 \leqq 0, \quad k = 1, \ldots, r,$$

amiből kapjuk, hogy $G_k(\mathbf{y}) \leqq -\vartheta_0<0$, $k=1, \ldots, r$, tehát a primál feladat szuperkonzisztens.

Fordítva, ha a primál feladat szuperkonzisztens, akkor van olyan $\mathbf{y}$, hogy $G_k(\mathbf{y})<0$, $k=1, \ldots, r$, és ez optimális megoldás is $\mathbf{f}=\mathbf{0}$ mellett. Ekkor tetszőleges

$(\mathbf{x}, \mathbf{z}) \in D$ esetén $\mathbf{cx}+\mathbf{dz}+F(\mathbf{x}, \mathbf{z}) \geqq 0$. Ha az

$$\mathbf{xA}+\mathbf{zB} = \mathbf{0}$$

$$\mathbf{z} \geqq \mathbf{0}$$

$$\zeta_k = 0 \Rightarrow \xi_i = 0, \quad i \in I_k, \quad k = 1, \ldots, r$$

$$\mathbf{cx}+\mathbf{dz}+F(\mathbf{x}, \mathbf{z}) \leqq 0$$

rendszernek nincs megoldása, akkor készen vagyunk. Ha van megoldása, akkor a 4.1. tétel miatt optimális megoldása is létezik, és erre a 7.3. tétel következményei miatt $\zeta_k G_k(\mathbf{y})=0$, $k=1, \ldots, r$. Mivel $G_k(\mathbf{y})<0$ minden k-ra, így $\zeta_k=0$, $k=1, \ldots, r$ és ekkor $\xi_i=0$, $i \in I_k$, $k=1, \ldots, r$, tehát a duál feladat reguláris.

Megjegyzés: Az *a)* állítás nem megfordítható, mert például legyen $\mathbf{A}=(1, 1)$, $\mathbf{B}=(1, 1)$, $\mathbf{f}=(1, 1)$, $\mathbf{c}=(0)$, $\mathbf{d}=(0)$, $p=q=2$, ekkor a primál feladat nem reguláris, mert az $\eta_1+\eta_2=0$, $\eta_1+\eta_2 \leqq 0$, $\eta_1+\eta_2 \geqq 0$ egyenlőtlenség-rendszernek például az $\mathbf{y}=(-1, 1)$ nem triviális megoldása, de a duál feladat *Slater-reguláris,* mivel az $x+z=1$, $z \geqq 0$, $z=0 \Rightarrow x=0$ rendszernek az $x=0$, $z=1$ olyan megoldása, hogy $z>0$.

9.2. TÉTEL. *a)* Legyen $P^* \neq \varnothing$, ekkor P^* akkor és csak akkor korlátos, ha a primál feladat reguláris,

b) Legyen $P \neq \varnothing$ és a primál feladat reguláris, ekkor $P^* \neq \varnothing$.

c) Legyen $D^* \neq \varnothing$, ekkor D^* akkor és csak akkor korlátos, ha a duál feladat reguláris.

d) Legyen $D \neq \varnothing$ és a duál feladat reguláris, akkor $D^* \neq \varnothing$.

Bizonyítás. a) Legyen μ a primál feladat optimális értéke, ekkor

$$\varnothing \neq P^* = \left\{ \mathbf{y} \,\middle|\, \begin{matrix} G_k(\mathbf{y}) \leqq 0, & k = 1, \ldots, r \\ \mathbf{fy}-\mu \geqq 0 \end{matrix} \right\}$$

akkor és csak akkor korlátos a 3.4. lemma miatt, ha a primál feladat reguláris.

b) Ha $P \neq \varnothing$ és a primál feladat reguláris, akkor a 9.1. tétel miatt a duál feladat *Slater-reguláris,* így $D \neq \varnothing$ és a 7.3. tétel miatt $P^* \neq \varnothing$.

c) Legyen $D^* \neq \varnothing$. Tegyük fel, hogy D^* nem korlátos, akkor van olyan $(\mathbf{x}^1, \mathbf{z}^1), \ldots, (\mathbf{x}^t, \mathbf{z}^t), \ldots \in D^*$ sorozat, hogy $\|(\mathbf{x}^t, \mathbf{z}^t)\| \to +\infty$, $t \to +\infty$ esetén, ahol

$$\|(\mathbf{x}^t, \mathbf{z}^t)\| = \sum_{i=1}^{n} |\xi_i^t| + \sum_{k=1}^{r} \zeta_k^t.$$

A t-edik elemre a feladat a következő:

$$\mathbf{x}^t\mathbf{A}+\mathbf{z}^t\mathbf{B} = \mathbf{f}$$

$$\mathbf{z}^t \geqq \mathbf{0}$$

$$\zeta_k^t = 0 \Rightarrow \xi_i^t = 0, \quad i \in I_k, \quad k = 1, \ldots, r$$

$$\mathbf{cx}^t+\mathbf{dz}^t+F(\mathbf{x}^t, \mathbf{z}^t) = v.$$

A normával való osztás után nyert sorozatot jelöljük $(\tilde{\mathbf{x}}^t, \tilde{\mathbf{z}}^t)$-vel,

$$\tilde{\mathbf{x}}^t = \frac{\mathbf{x}^t}{\|(\mathbf{x}^t, \mathbf{z}^t)\|}, \quad \tilde{\mathbf{z}}^t = \frac{\mathbf{z}^t}{\|(\mathbf{x}^t, \mathbf{z}^t)\|}, \quad \tilde{\mathbf{f}}^t = \frac{\mathbf{f}}{\|(\mathbf{x}^t, \mathbf{z}^t)\|}, \quad \tilde{v}^t = \frac{v}{\|(\mathbf{x}^t, \mathbf{z}^t)\|}.$$

Így a következő feladat adódik:

$$\tilde{\mathbf{x}}^t\mathbf{A} + \tilde{\mathbf{z}}^t\mathbf{B} = \tilde{\mathbf{f}}^t$$

$$\tilde{\mathbf{z}}^t \geqq \mathbf{0}$$

$$\tilde{\zeta}_k^t = 0 \Rightarrow \tilde{\xi}_i^t = 0, \quad i \in I_k, \quad k = 1, \ldots, r$$

$$\mathbf{c}\tilde{\mathbf{x}}^t + \mathbf{d}\tilde{\mathbf{z}}^t + F(\tilde{\mathbf{x}}^t, \tilde{\mathbf{z}}^t) = \tilde{v}^t.$$

Az $(\tilde{\mathbf{x}}^t, \tilde{\mathbf{z}}^t)$ sorozat korlátos, így kiválasztható konvergens részsorozat. Jelöljük azt is ugyanígy. Legyen $(\tilde{\mathbf{x}}, \tilde{\mathbf{z}})$ a limeszpont, és ekkor határátmenettel nyerjük:

$$\tilde{\mathbf{x}}\mathbf{A} + \tilde{\mathbf{z}}\mathbf{B} = 0$$

$$\tilde{\mathbf{z}} \geqq \mathbf{0}, \quad \|(\tilde{\mathbf{x}}, \tilde{\mathbf{z}})\| = 1$$

$$\tilde{\zeta}_k = 0 \Rightarrow \tilde{\xi}_i = 0, \quad i \in I_k, \quad k = 1, \ldots, r$$

$$\mathbf{c}\tilde{\mathbf{x}} + \mathbf{d}\tilde{\mathbf{z}} + F(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}) = 0.$$

Ezek az összefüggések nyilván igazak $(\tilde{\mathbf{x}}, \tilde{\mathbf{z}})$-re, egyedül az igényel bizonyítást, hogy $\tilde{\zeta}_k = 0 \Rightarrow \tilde{\xi}_i = 0$, $i \in I_k$, $k = 1, \ldots, r$. Tegyük fel, hogy ez nem áll fenn, ez azt jelenti, hogy valamely k és $i \in I_k$ esetén $\zeta_k^t \to 0$, de $\xi_i^t \not\to 0$, miközben $t \to +\infty$, ez pedig azt jelentené, hogy ekkor $\mathbf{c}\mathbf{x}^t + \mathbf{d}\mathbf{z}^t + F(\mathbf{x}^t, \mathbf{z}^t) \to +\infty$. Ez ellentmond annak, hogy az $(\mathbf{x}^t, \mathbf{z}^t)$ sorozat optimális megoldások sorozata.

Tehát ekkor a duál feladat nem reguláris, mivel $(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}) \neq \mathbf{0}$.

Fordítva, ha a duál feladat nem reguláris, akkor létezik $(\tilde{\mathbf{x}}, \tilde{\mathbf{z}})$ vektor úgy, hogy

$$\tilde{\mathbf{x}}\mathbf{A} + \tilde{\mathbf{z}}\mathbf{B} = \mathbf{0}$$

$$\tilde{\mathbf{z}} \geqq \mathbf{0}, \quad \|(\tilde{\mathbf{x}}, \tilde{\mathbf{z}})\| = 1$$

$$\tilde{\zeta}_k = 0 \Rightarrow \tilde{\xi}_i = 0, \quad i \in I_k, \quad k = 1, \ldots, r$$

$$\mathbf{c}\tilde{\mathbf{x}} + \mathbf{d}\tilde{\mathbf{z}} + F(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}) \leqq 0.$$

Ekkor tetszőleges $\vartheta \geqq 0$ és $(\mathbf{x}, \mathbf{z}) \in D^*$ esetén a 3.5. lemma miatt $(\mathbf{x} + \vartheta\tilde{\mathbf{x}}, \mathbf{z} + \vartheta\tilde{\mathbf{z}}) \in D$, továbbá F szubadditivitása miatt

$$v \leqq \mathbf{c}(\mathbf{x} + \vartheta\tilde{\mathbf{x}}) + \mathbf{d}(\mathbf{z} + \vartheta\tilde{\mathbf{z}}) + F(\mathbf{x} + \vartheta\tilde{\mathbf{x}}, \mathbf{z} + \vartheta\tilde{\mathbf{z}}) \leqq \mathbf{c}\mathbf{x} + \mathbf{d}\mathbf{z} + F(\mathbf{x}, \mathbf{z}) +$$

$$+ \vartheta\left(\mathbf{c}\tilde{\mathbf{x}} + \mathbf{d}\tilde{\mathbf{z}} + F(\tilde{\mathbf{x}}, \tilde{\mathbf{z}})\right) \leqq \mathbf{c}\mathbf{x} + \mathbf{d}\mathbf{z} + F(\mathbf{x}, \mathbf{z}) = v.$$

Tehát $(\mathbf{x} + \vartheta\tilde{\mathbf{x}}, \mathbf{z} + \vartheta\tilde{\mathbf{z}}) \in D^*$ és így D^* nem korlátos.

d) Ha a duál feladat reguláris, akkor a 9.1. tétel miatt a primál feladat szuperkonzisztens, ami egyben azt is jelenti, hogy kielégíti a *Slater-regularitási feltételt*. Mivel P sem és D sem üres, így $\mu = v =$ véges és ekkor a 4.1. tétel miatt a duál feladatnak is létezik optimális megoldása.

KÖVETKEZMÉNY. Ha az l_p programozási feladatpár reguláris, akkor a primál és a duál feladatnak is van optimális megoldása, $\mu = \nu$ és az optimális halmazok korlátosak.

Bizonyítás. Ha reguláris a feladatpár, akkor a 9.1. tétel miatt a primál feladat szuperkonzisztens, a duál feladat *Slater-reguláris,* így a 9.2. tétel *b)* és *d)* pontja miatt $P^* \neq \emptyset$, $D^* \neq \emptyset$. A 7.2. tétel miatt $\mu = \nu$, továbbá a 9.2. tétel *a)* és *c)* pontja miatt P^* és D^* is korlátos.

Jelöljük a továbbiakban [**A**, **B**, **c**, **d**, **f**]-fel az ezekhez a mátrixokhoz, illetve vektorokhoz tartozó l_p programozási feladatpárt.

9.3. LEMMA. A következő két halmaz bármely rögzített $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{c}}, \hat{\mathbf{d}}, \hat{\mathbf{f}}$ esetén (amelyek azonos dimenziósak, mint **A**, **B**, **c**, **d**, **f**) nyílt:

$$\{\tau | [\mathbf{A}+\tau\hat{\mathbf{A}}, \mathbf{B}+\tau\hat{\mathbf{B}}, \mathbf{c}+\tau\hat{\mathbf{c}}, \mathbf{d}+\tau\hat{\mathbf{d}}, \mathbf{f}+\tau\hat{\mathbf{f}}] \text{ primál feladata reguláris}\},$$

$$\{\tau | [\mathbf{A}+\tau\hat{\mathbf{A}}, \mathbf{B}+\tau\hat{\mathbf{B}}, \mathbf{c}+\tau\hat{\mathbf{c}}, \mathbf{d}+\tau\hat{\mathbf{d}}, \mathbf{f}+\tau\hat{\mathbf{f}}] \text{ duál feladata reguláris}\}.$$

Bizonyítás. Csak az első halmaz esetében bizonyítjuk, a másik halmazra hasonlóan bizonyítható. Az egyszerűség kedvéért tegyük fel, hogy $\tau = 0$-ra a primál feladat reguláris. Indirekt tegyük fel, hogy nincs $\tau_0 > 0$ úgy, hogy $\tau \in [0, \tau_0)$ esetén a primál feladat reguláris, akkor van $\tau_1, \ldots, \tau_t, \ldots \to 0$, $\tau_t \geqq 0$ és $\{\mathbf{y}_t\}_{t=1}^{\infty}$ sorozat úgy, hogy

$$(\mathbf{A}+\tau_t\hat{\mathbf{A}})\mathbf{y}_t = 0$$

$$(\mathbf{B}+\tau_t\hat{\mathbf{B}})\mathbf{y}_t \leqq 0, \quad \mathbf{y}_t \neq 0.$$

$$(\mathbf{f}+\tau_t\hat{\mathbf{f}})\mathbf{y}_t \geqq 0$$

Legyen $\tilde{\mathbf{y}}_t = \frac{\mathbf{y}_t}{\|\mathbf{y}_t\|}$, így

$$(\mathbf{A}+\tau_t\hat{\mathbf{A}})\tilde{\mathbf{y}}_t = 0$$

$$(\mathbf{B}+\tau_t\hat{\mathbf{B}})\tilde{\mathbf{y}}_t \leqq 0, \quad \|\tilde{\mathbf{y}}_t\| = 1.$$

$$(\mathbf{f}+\tau_t\hat{\mathbf{f}})\tilde{\mathbf{y}}_t \geqq 0$$

Az $\tilde{\mathbf{y}}_t$ sorozat korlátos, kiválasztható belőle konvergens részsorozat. Legyen $\tilde{\mathbf{y}}$ a limeszpont és erre határátmenettel nyerjük, hogy

$$\mathbf{A}\tilde{\mathbf{y}} = \mathbf{0}$$

$$\mathbf{B}\tilde{\mathbf{y}} \leqq \mathbf{0}, \quad \|\tilde{\mathbf{y}}\| = 1$$

$$\mathbf{f}\tilde{\mathbf{y}} \geqq 0$$

Ez ellentmond annak, hogy $\tau = 0$ esetén a primál feladat reguláris.

10. Érzékenységvizsgálat

Legyenek $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{c}}, \hat{\mathbf{d}}, \hat{\mathbf{f}}$ azonos dimenziósak, mint $\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{d}, \mathbf{f}$ és vizsgáljuk azt, hogy az $[\mathbf{A}+\tau\hat{\mathbf{A}}, \mathbf{B}+\tau\hat{\mathbf{B}}, \mathbf{c}+\tau\hat{\mathbf{c}}, \mathbf{d}+\tau\hat{\mathbf{d}}, \mathbf{f}+\tau\hat{\mathbf{f}}]$ feladat hogyan viselkedik $\tau=0$ közelében, azaz a feladatpár közös értéke hogyan változik τ változásával.

Jelöljük $P_\tau, D_\tau, P_\tau^*, D_\tau^*$-gal a fenti feladatokhoz tartozó megengedett és optimális halmazokat.

10.1. TÉTEL. *a)* Az $[\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{d}, \mathbf{f}]$ l_p programozási feladatpár regularitásának szükséges és elégséges feltétele az, hogy tetszőleges $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{c}}, \hat{\mathbf{d}}, \hat{\mathbf{f}}$-höz létezzen $\tau_0>0$ úgy, hogy $\tau\in[0, \tau_0)$ esetén az $[\mathbf{A}+\tau\hat{\mathbf{A}}, \mathbf{B}+\tau\hat{\mathbf{B}}, \mathbf{c}+\tau\hat{\mathbf{c}}, \mathbf{d}+\tau\hat{\mathbf{d}}, \mathbf{f}+\tau\hat{\mathbf{f}}]$-hoz tartozó P_τ és D_τ halmazok ne legyenek üresek.

b) Ha az $[\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{d}, \mathbf{f}]$ l_p programozási feladatpár reguláris, akkor tetszőleges $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{c}}, \hat{\mathbf{d}}, \hat{\mathbf{f}}$-höz van olyan $\tau_0>0$, hogy $\tau\in[0, \tau_0)$ esetén $P_\tau^*\neq\emptyset$ és $D_\tau^*\neq\emptyset$, továbbá $\mu(\tau)$ közös értékre fennáll a következő összefüggés:

$$\lim_{\tau\to 0}\frac{\mu(\tau)-\mu(0)}{\tau}=\max_{\mathbf{y}\in P^*}\min_{(\mathbf{x},\mathbf{z})\in D^*}\{\hat{\mathbf{c}}\mathbf{x}+\hat{\mathbf{d}}\mathbf{z}+\hat{\mathbf{f}}\mathbf{y}-(\mathbf{x}\hat{\mathbf{A}}+\mathbf{z}\hat{\mathbf{B}})\mathbf{y}\}.$$

Bizonyítás. a) Először azt bizonyítjuk, hogy $[\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{d}, \mathbf{f}]$ regularitása szükséges feltétel. Tegyük fel indirekt, hogy a primál feladat nem reguláris, akkor van olyan $\mathbf{y}\neq\mathbf{0}$, hogy $\mathbf{A}\mathbf{y}=\mathbf{0}, \mathbf{B}\mathbf{y}\leqq\mathbf{0}, \mathbf{f}\mathbf{y}\geqq 0$. Mivel tetszőleges $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{c}}, \hat{\mathbf{d}}, \hat{\mathbf{f}}$-höz van olyan $\tau_0>0$, hogy $\tau\in[0, \tau_0)$ esetén $P_\tau, D_\tau\neq\emptyset$, így $\hat{\mathbf{A}}=\mathbf{0}, \hat{\mathbf{B}}=\mathbf{0}, \hat{\mathbf{c}}=\mathbf{0}, \hat{\mathbf{d}}=-1, \hat{\mathbf{f}}$ olyan, hogy $\hat{\mathbf{f}}\mathbf{y}>0$-hoz is létezik ilyen τ_0. Ekkor $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{c}}, \hat{\mathbf{d}}, \hat{\mathbf{f}}$ megválasztása miatt

$$\begin{aligned}(\mathbf{A}+\tau\hat{\mathbf{A}})\mathbf{y}&=0\\ (\mathbf{B}+\tau\hat{\mathbf{B}})\mathbf{y}&\leqq 0\\ (\mathbf{f}+\tau\hat{\mathbf{f}})\mathbf{y}&>0.\end{aligned}\tag{10.1}$$

Legyen $\hat{\mathbf{y}}\in P_\tau$, akkor a 3.3. lemma miatt $\hat{\mathbf{y}}+\vartheta\mathbf{y}\in P_\tau$ és (10.1) miatt $(\mathbf{f}+\tau\hat{\mathbf{f}})(\hat{\mathbf{y}}+\vartheta\mathbf{y})\to$ $\to+\infty$, ha $\vartheta\to+\infty$, ami ellentmond annak, hogy $D_\tau\neq\emptyset$, így a primál feladat reguláris.

Mivel $\tau\in[0, \tau_0)$ esetén $P_\tau\neq\emptyset$, így

$$\sum_{i\in I_k}\frac{1}{p_i}|(\mathbf{a}_i+\tau\mathbf{a}_i)\mathbf{y}-\gamma_i-\tau\hat{\gamma}_i|^{p_i}+(\mathbf{b}_k+\tau\hat{\mathbf{b}}_k)\mathbf{y}-\delta_k-\tau\hat{\delta}_k\leqq 0,\quad k=1, \ldots, r.$$

Ezt átalakítva:

$$\sum_{i\in I_k}\frac{1}{p_i}|\mathbf{a}_i\mathbf{y}-\gamma_i|^{p_i}+\mathbf{b}_k\mathbf{y}-\delta_k\leqq-\tau<0,\quad k=1, \ldots, r.$$

Ez azt jelenti, hogy a primál feladat szuperkonzisztens és így a 9.1. tétel miatt a duál feladat reguláris. Mivel láttuk, hogy a primál feladat is reguláris, így a feladatpár reguláris.

Fordítva, ha az [**A**, **B**, **c**, **d**, **f**] feladat reguláris, akkor a 9.3. lemma miatt van olyan $\tau_0>0$, hogy $\tau\in[0, \tau_0)$ esetén $[\mathbf{A}+\tau\hat{\mathbf{A}}, \mathbf{B}+\tau\hat{\mathbf{B}}, \mathbf{c}+\tau\hat{\mathbf{c}}, \mathbf{d}+\tau\hat{\mathbf{d}}, \mathbf{f}+\tau\hat{\mathbf{f}}]$ primál és duál feladata is reguláris, továbbá a 9.2. tétel miatt ekkor $P_\tau, D_\tau \neq \emptyset$.

b) Ha [**A**, **B**, **c**, **d**, **f**] reguláris, akkor a fentiek miatt létezik $\hat{\tau}>0$ úgy, hogy $\tau\in[0, \hat{\tau})$ esetén $P_\tau, D_\tau \neq \emptyset$, sőt a 9.2. tételből az is következik, hogy $P_\tau^*, D_\tau^* \neq \emptyset$ és korlátos.

Megmutatjuk, hogy létezik olyan $0<\tau_0\leqq\hat{\tau}$ úgy, hogy a $[0, \tau_0)$ intervallumba eső τ estén P_τ^*, D_τ^* egyenletesen korlátos.

Tegyük fel indirekt, hogy nem létezik ilyen τ_0, ekkor van olyan $\tau_1, \ldots, \tau_t, \ldots \to 0$, $\tau_t \geqq 0$ sorozat és $\mathbf{y}^1, \ldots, \mathbf{y}^t, \ldots$, valamint $(\mathbf{x}^1, \mathbf{z}^1), \ldots, (\mathbf{x}^t, \mathbf{z}^t), \ldots$ optimális megoldások sorozata, hogy a következő három állítás közül pontosan egy teljesül:

$$\|(\mathbf{x}^t, \mathbf{z}^t)\| \to +\infty \quad \text{és} \quad \|\mathbf{y}^t\| \text{ korlátos},$$

$$\|(\mathbf{x}^t, \mathbf{z}^t)\| \text{ korlátos} \quad \text{és} \quad \|\mathbf{y}^t\| \to +\infty,$$

$$\|(\mathbf{x}^t, \mathbf{z}^t)\| \to +\infty \quad \text{és} \quad \|\mathbf{y}^t\| \to +\infty.$$

Normáljuk a megoldássorozatokat

$$\tilde{\mathbf{y}}^t = \frac{\mathbf{y}^t}{\|\mathbf{y}^t\|}, \quad (\tilde{\mathbf{x}}^t, \tilde{\mathbf{z}}^t) = \frac{(\mathbf{x}^t, \mathbf{z}^t)}{\|(\mathbf{x}^t, \mathbf{z}^t)\|}.$$

Az így nyert sorozatok korlátosak, válasszunk ki ezekből konvergens részsorozatokat, és jelöljük azokat is ugyanígy. Legyen $\tilde{\mathbf{y}}$, $(\tilde{\mathbf{x}}, \tilde{\mathbf{z}})$ a limeszpont, azaz $\tilde{\mathbf{y}}^t \to \tilde{\mathbf{y}}$, $(\tilde{\mathbf{x}}^t, \tilde{\mathbf{z}}^t) \to (\tilde{\mathbf{x}}, \tilde{\mathbf{z}})$, ha $t \to \infty$. Ismert, hogy $\mathbf{y}^t$, $(\mathbf{x}^t, \mathbf{z}^t)$ és a hozzájuk tartozó τ_t-re fennállnak a következők:

$$\mathbf{x}^t(\mathbf{A}+\tau_t\hat{\mathbf{A}})+\mathbf{z}^t(\mathbf{B}+\tau_t\hat{\mathbf{B}}) = \mathbf{f}+\tau_t\hat{\mathbf{f}}$$

$$\mathbf{z}^t \geqq \mathbf{0} \tag{10.2}$$

$$\zeta_k^t = 0 \Rightarrow \xi_i^t = 0, \quad i\in I_k, \quad k = 1, \ldots, r$$

$$\mu(\tau_t) = \mu_t = (\mathbf{f}+\tau_t\hat{\mathbf{f}})\mathbf{y}^t = \mathbf{c}\mathbf{x}^t+\mathbf{d}\mathbf{z}^t+\tau_t(\hat{\mathbf{c}}\mathbf{x}^t+\hat{\mathbf{d}}\mathbf{z}^t)+F(\mathbf{x}^t, \mathbf{z}^t) \tag{10.3}$$

$$\sum_{i\in I_k}\frac{1}{p_i}|(\mathbf{a}_i+\tau_t\hat{\mathbf{a}}_i)\mathbf{y}^t-\gamma_i-\tau_t\hat{\gamma}_i|^{p_i}+(\mathbf{b}_k+\tau_t\hat{\mathbf{b}}_k)\mathbf{y}^t-\delta_k-\tau_t\hat{\delta}_k \leqq 0, \quad k = 1, \ldots, r. \tag{10.4}$$

Továbbá (10.4)-ből következik

$$(\mathbf{B}+\tau_t\hat{\mathbf{B}})\mathbf{y}^t \leqq \mathbf{d}+\tau_t\hat{\mathbf{d}}. \tag{10.5}$$

Végezzük el (10.2) és (10.3)-ban az $\|(\mathbf{x}^t, \mathbf{z}^t)\|$-val való osztást.

$$\tilde{\mathbf{x}}(\mathbf{A}+\tau_t\hat{\mathbf{A}})+\tilde{\mathbf{z}}^t(\mathbf{B}+\tau_t\hat{\mathbf{B}}) = \frac{1}{\|(\mathbf{x}^t, \mathbf{z}^t)\|}(\mathbf{f}+\tau_t\hat{\mathbf{f}})$$

$$\tilde{\mathbf{z}}^t \geqq \mathbf{0}$$

$$\tilde{\zeta}_k^t = 0 \Rightarrow \tilde{\xi}_i^t = 0, \quad i\in I_k, \quad k = 1, \ldots, r$$

$$\|(\tilde{\mathbf{x}}^t, \tilde{\mathbf{z}}^t)\| = 1$$

$$(\mathbf{c}+\tau_t\hat{\mathbf{c}})\tilde{\mathbf{x}}^t+(\mathbf{d}+\tau_t\hat{\mathbf{d}})\tilde{\mathbf{z}}^t+F(\tilde{\mathbf{x}}^t, \tilde{\mathbf{z}}^t) = \frac{1}{\|(\mathbf{x}^t, \mathbf{z}^t)\|}(\mathbf{f}+\tau_t\hat{\mathbf{f}})\mathbf{y}^t.$$

Ha $\|(\mathbf{x}^t, \mathbf{z}^t)\| \to +\infty$ és $\|\mathbf{y}^t\|$ korlátos, akkor határátmenettel nyerjük, hogy

$$\tilde{\mathbf{x}}\mathbf{A}+\tilde{\mathbf{z}}\mathbf{B} = \mathbf{0}$$

$$\tilde{\mathbf{z}} \geqq \mathbf{0}$$

$$\|(\tilde{\mathbf{x}}, \tilde{\mathbf{z}})\| = 1$$

$$\mathbf{c}\tilde{\mathbf{x}}+\mathbf{d}\tilde{\mathbf{z}}+F(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}) = 0,$$

Mivel $\|\mathbf{y}^t\|$ korlátos, így $\tilde{\zeta}_k=0 \Rightarrow \tilde{\xi}_i=0$, $i\in I_k$, $k=1, \ldots, r$, mert különben a célfüggvények egyenlőségét felíró feltétel esetén a bal oldal $+\infty$-hez tartana, miközben a jobb oldal 0-hoz. Így ellentmondásba kerültünk a duál feladat regularitásával, tehát ez az eset nem lehetséges.

Ha $\|(\mathbf{x}^t, \mathbf{z}^t)\|$ korlátos és $\|\mathbf{y}^t\| \to +\infty$, akkor hasonló technikával, limesz képzéssel (10.3) és (10.5)-ből következik, hogy

$$\mathbf{B}\tilde{\mathbf{y}} \leqq \mathbf{0},$$

$$\mathbf{f}\tilde{\mathbf{y}} = 0,$$

$$\|\tilde{\mathbf{y}}\| = 1,$$

továbbá (10.4) miatt

$$\frac{1}{p_i}|(\mathbf{a}_i+\tau_t\hat{\mathbf{a}}_i)\mathbf{y}^t-\gamma_i-\tau_t\hat{\gamma}_i|^{p_i}+(\mathbf{b}_k+\tau_t\hat{\mathbf{b}}_k)\mathbf{y}^t-\delta_k-\tau_t\hat{\delta}_k \leqq 0, \quad i\in I_k, \quad k = 1, \ldots, r.$$

Ezt átrendezve

$$|(\mathbf{a}_i+\tau_t\hat{\mathbf{a}}_i)\mathbf{y}^t-\gamma_i-\tau_t\hat{\gamma}_i| \leqq \{p_i[-(\mathbf{b}_k+\tau_t\hat{\mathbf{b}}_k)\mathbf{y}^t+\delta_k+\tau_t\hat{\delta}_k]\}^{\frac{1}{p_i}}, \quad i\in I_k, \quad k = 1, \ldots, r.$$

Mindkét oldalt $\|\mathbf{y}^t\|$-val osztva, határátmenetet képezve nyerjük, hogy $|\mathbf{a}_i\tilde{\mathbf{y}}| \leqq 0$, $i\in I_k$, $k=1, \ldots, r$. Így $\mathbf{A}\tilde{\mathbf{y}}=\mathbf{0}$ és a primál feladat regularitásával ellentmondásba jutottunk így, tehát ez az eset sem lehetséges.

Ha $\|\mathbf{y}^t\| \to +\infty$ és $\|(\mathbf{x}^t, \mathbf{z}^t)\| \to +\infty$, akkor hasonlóan az előző két esethez határátmenettel nyerjük (10.2), (10.4), (10.5)-ből, hogy

$$\tilde{\mathbf{x}}\mathbf{A}+\tilde{\mathbf{z}}\mathbf{B} = \mathbf{0} \qquad \mathbf{A}\tilde{\mathbf{y}} = \mathbf{0}$$

$$\tilde{\mathbf{z}} \geqq \mathbf{0} \qquad \mathbf{B}\tilde{\mathbf{y}} \leqq \mathbf{0}$$

$$\|(\tilde{\mathbf{x}}, \tilde{\mathbf{z}})\| = 1 \qquad \|\tilde{\mathbf{y}}\| = 1$$

valamint (10.3)-ból

$$\|\mathbf{y}^t\|(\mathbf{f}+\tau_t\hat{\mathbf{f}})\tilde{\mathbf{y}}^t = \|(\mathbf{x}^t, \mathbf{z}^t)\|\left((\mathbf{c}+\tau_t\hat{\mathbf{c}})\tilde{\mathbf{x}}^t+(\mathbf{d}+\tau_t\hat{\mathbf{d}})\tilde{\mathbf{z}}^t+F(\tilde{\mathbf{x}}^t, \tilde{\mathbf{z}}^t)\right).$$

A primál feladat regularitása miatt $\mathbf{f}\tilde{\mathbf{y}}<0$, ebből következik, hogy $\tilde{\zeta}_k=0 \Rightarrow \tilde{\xi}_i=0$, $i\in I_k$, $k=1, \ldots, r$, mert ha ez nem állna fenn, akkor az egyenlet jobb oldala $+\infty$-hez tartana, miközben a bal oldal $-\infty$-hez.

A duál feladat regularitása miatt $\mathbf{c}\tilde{\mathbf{x}}+\mathbf{d}\tilde{\mathbf{z}}+F(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}) > 0$, valamint a primál feladat regularitása miatt $\mathbf{f}\tilde{\mathbf{y}} < 0$, s így létezik τ_t úgy, hogy

$$(\mathbf{c}+\tau_t\hat{\mathbf{c}})\tilde{\mathbf{x}}^t+(\mathbf{d}+\tau_t\hat{\mathbf{d}})\tilde{\mathbf{z}}^t+F(\tilde{\mathbf{x}}^t, \tilde{\mathbf{z}}^t) > 0, \quad (\mathbf{f}+\tau_t\hat{\mathbf{f}})\tilde{\mathbf{y}}^t < 0.$$

Erre a t-re nem áll fenn a célfüggvényekre vonatkozó egyenlet, tehát az optimális megoldás halmazok egyenletesen korlátosak.

Most belátjuk, hogy ha $\{\mathbf{y}^t\}$ és $\{(\mathbf{x}^t, \mathbf{z}^t)\}$ optimális megoldások sorozata $\tau_t \to 0$ esetén, akkor a megfelelő torlódási pontok P^*, illetve D^*-ban vannak.

Legyen $\mathbf{y}$ és $(\mathbf{x}, \mathbf{z})$ a két torlódási pont. Ezek léteznek, mivel $\tau \in [0, \tau_0)$ esetén P_τ^*, D_τ^* egyenletesen korlátos. Létezik hozzájuk konvergáló (ugyanígy jelölt) $\mathbf{y}^t$, $(\mathbf{x}^t, \mathbf{z}^t)$ részsorozat, vagyis

$$\mathbf{y}^t \to \mathbf{y}, \quad (\mathbf{x}^t, \mathbf{z}^t) \to (\mathbf{x}, \mathbf{z}), \quad t \to +\infty.$$

Mivel τ_τ esetén fennáll (10.2), (10.3), (10.4), (10.5), ezekből határátmenettel nyerjük, hogy:

$$\mathbf{xA} + \mathbf{zB} = \mathbf{f}$$

$$\mathbf{z} \geqq \mathbf{0} \qquad \sum_{i \in I_k} \frac{1}{p_i} |\mathbf{a}_i \mathbf{y} - \gamma_i|^{p_i} + \mathbf{b}_k \mathbf{y} - \delta_k \leqq 0, \quad k = 1, \ldots, r.$$

$$\mathbf{fy} = \mathbf{cx} + \mathbf{dz} + F(\mathbf{x}, \mathbf{z})$$

Ekkor $\zeta_k = 0 \Rightarrow \xi_i = 0$, $i \in I_k$, $k = 1, \ldots, r$ mivel különben a célfüggvényekre vonatkozó egyenlet jobb oldala $+\infty$-hez tartana. Tehát a fő lemma miatt $\mathbf{y} \in P^*$, $(\mathbf{x}, \mathbf{z}) \in D^*$.

Végül a duál feladat *Lagrange függvényének* nyeregpont-egyenlőtlenségét használjuk fel az érzékenységi formula meghatározására.

A Ψ *Lagrange-függvényt* a 8.3. definícióban adtuk meg. Jelöljük Ψ_τ-val a $\tau \in [0, \tau_0)$-hoz tartozó *Lagrange-függvényt*. Legyenek $(\mathbf{x}_0, \mathbf{z}_0)$, $\mathbf{y}_0$ az eredeti, $(\mathbf{x}_\tau, \mathbf{z}_\tau)$, $\mathbf{y}_\tau$ a τ-hoz tartozó feladatok optimális megoldásai. Ekkor fennáll a következő két nyeregpont-egyenlőtlenség.

$$\Psi_\tau(\mathbf{x}_\tau, \mathbf{z}_\tau, \mathbf{y}_0) \leqq \mu(\tau) \leqq \Psi_\tau(\mathbf{x}_0, \mathbf{z}_0, \mathbf{y}_\tau)$$

$$\Psi_0(\mathbf{x}_0, \mathbf{z}_0, \mathbf{y}_\tau) \leqq \mu(0) \leqq \Psi_0(\mathbf{x}_\tau, \mathbf{z}_\tau, \mathbf{y}_0).$$

Az elsőből a másodikat kivonva és τ-val osztva

$$\hat{\mathbf{c}}\mathbf{x}_\tau + \hat{\mathbf{d}}\mathbf{z}_\tau + \hat{\mathbf{f}}\mathbf{y}_0 - (\mathbf{x}_\tau \hat{\mathbf{A}} + \mathbf{z}_\tau \hat{\mathbf{B}})\mathbf{y}_0 \leqq \frac{\mu(\tau) - \mu(0)}{\tau} \leqq \hat{\mathbf{c}}\mathbf{x}_0 + \hat{\mathbf{d}}\mathbf{z}_0 + \hat{\mathbf{f}}\mathbf{y}_\tau - (\mathbf{x}_0 \hat{\mathbf{A}} + \mathbf{z}_0 \hat{\mathbf{B}})\mathbf{y}_\tau.$$

Ez fennáll tetszőleges $\mathbf{y}_0 \in P^*$ és $(\mathbf{x}_0, \mathbf{z}_0) \in D^*$ esetén, így

$$\max_{\mathbf{y} \in P^*} [\hat{\mathbf{c}}\mathbf{x}_\tau + \hat{\mathbf{d}}\mathbf{z}_\tau + \hat{\mathbf{f}}\mathbf{y} - (\mathbf{x}_\tau \hat{\mathbf{A}} + \mathbf{z}_\tau \hat{\mathbf{B}})\mathbf{y}] \leqq \frac{\mu(\tau) - \mu(0)}{\tau} \leqq$$

$$\leqq \min_{(\mathbf{z}, \mathbf{x}) \in D^*} [\hat{\mathbf{c}}\mathbf{x} + \hat{\mathbf{d}}\mathbf{z} + \hat{\mathbf{f}}\mathbf{y}_\tau - (\mathbf{x}\hat{\mathbf{A}} + \mathbf{z}\hat{\mathbf{B}})\mathbf{y}_\tau].$$

Mivel $\lim_{\tau \to 0} \mathbf{y}_\tau \in P^*$ és $\lim_{\tau \to 0} (\mathbf{x}_\tau, \mathbf{z}_\tau) \in D^*$, így tetszőleges $\varepsilon > 0$-hoz létezik τ úgy, hogy

$$\min_{(\mathbf{x}, \mathbf{z}) \in D^*} [\hat{\mathbf{c}}\mathbf{x} + \hat{\mathbf{d}}\mathbf{z} + \hat{\mathbf{f}}\mathbf{y}_\tau - (\mathbf{x}\hat{\mathbf{A}} + \mathbf{z}\hat{\mathbf{B}})\mathbf{y}_\tau] \leqq \max_{\mathbf{y} \in P^*} \min_{(\mathbf{x}, \mathbf{z}) \in D^*} [\hat{\mathbf{c}}\mathbf{x} + \hat{\mathbf{d}}\mathbf{z} + \hat{\mathbf{f}}\mathbf{y} - (\mathbf{x}\hat{\mathbf{A}} + \mathbf{z}\mathbf{B})\mathbf{y}] + \varepsilon,$$

valamint

$$\max_{\mathbf{y} \in P^*} [\hat{\mathbf{c}}\mathbf{x}_\tau + \hat{\mathbf{d}}\mathbf{z}_\tau + \hat{\mathbf{f}}\mathbf{y} - (\mathbf{x}_\tau \hat{\mathbf{A}} + \mathbf{z}_\tau \hat{\mathbf{B}})\mathbf{y}] \geqq \min_{(\mathbf{x}, \mathbf{z}) \in D^*} \max_{\mathbf{y} \in P^*} [\hat{\mathbf{c}}\mathbf{x} + \hat{\mathbf{d}}\mathbf{z} + \hat{\mathbf{f}}\mathbf{y} - (\mathbf{x}\hat{\mathbf{A}} + \mathbf{z}\hat{\mathbf{B}})\mathbf{y}] - \varepsilon.$$

Az utóbbiakból nyerjük, hogy

$$-\varepsilon \leqq \frac{\mu(\tau)-\mu(0)}{\tau} - \max_{\mathbf{y}\in P^*} \min_{(\mathbf{x},\mathbf{z})\in D^*} [\hat{\mathbf{c}}\mathbf{x}+\hat{\mathbf{d}}\mathbf{z}+\hat{\mathbf{f}}\mathbf{y}-(\mathbf{x}\hat{\mathbf{A}}+\mathbf{z}\hat{\mathbf{B}})\mathbf{y}] \leqq \varepsilon.$$

Ebből pedig $\tau\to 0$ limeszképzéssel nyerjük a kívánt formulát:

$$\lim_{\tau\to 0} \frac{\mu(\tau)-\mu(0)}{\tau} = \max_{\mathbf{y}\in P^*} \min_{(\mathbf{x},\mathbf{z})\in D^*} [\hat{\mathbf{c}}\mathbf{x}+\hat{\mathbf{d}}\mathbf{z}+\hat{\mathbf{f}}\mathbf{y}-(\mathbf{x}\hat{\mathbf{A}}+\mathbf{z}\hat{\mathbf{B}})\mathbf{y}].$$

Tehát így tételünket teljes egészében beláttuk.

11. Speciális esetek

Ebben a fejezetben megmutatjuk, hogy néhány matematikai programozási feladattípus milyen módon adódik az l_p programozás speciális eseteként. Bemutatjuk az eddig ismert duál feladatok és az l_p programozás segítségével nyert duál feladatok kapcsolatát is.

Lineáris programozás.

A lineáris programozás primál és duál feladata a következő:

$$\begin{array}{ll} \max \mathbf{fy} & \min \mathbf{dz} \\ \mathbf{By} \leqq \mathbf{d} & \mathbf{zB} = \mathbf{f} \\ & \mathbf{z} \geqq \mathbf{0} \end{array}$$

Ez a feladatpár pedig egy l_p programozási feladatpár, ahol $I_k=\varnothing$, $k=1, \ldots, r$, $G_k(\mathbf{y})=\mathbf{b}_k\mathbf{y}-\delta_k$, $k=1, \ldots, r$. Látható, hogy a lineáris programozásban bemutatott duál feladat megegyezik az l_p programozásban definiált duál feladattal.

Az l_p programozási ismereteink alapján közvetlenül adódik a lineáris programozás dualitás tétele:

Ha $P\neq\varnothing$ és $D\neq\varnothing$, akkor $\mu=\nu=$véges, továbbá mindkét feladatnak van optimális megoldása. ($P^*\neq\varnothing$, $D^*\neq\varnothing$.)

Ez a fő lemma, a 4.1. tétel és a 7.3. tétel közvetlen következménye.

Kvadratikus programozás.

A kvadratikus feltételes kvadratikus programozási feladatról mutatjuk meg, hogy az l_p programozás speciális esete.

A kvadratikus programozási feladat:

$$\inf R_0(\mathbf{y})$$

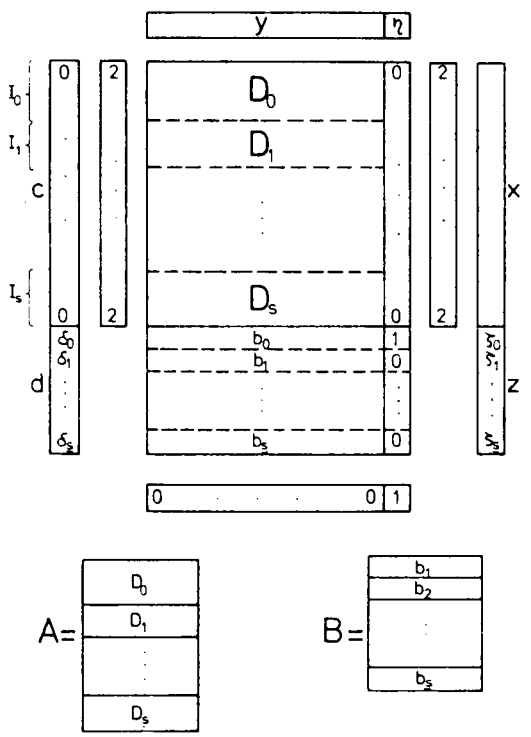
$$R_k(\mathbf{y}) \leqq 0, \quad k = 1, \ldots, s,$$

ahol $R_k(\mathbf{y})=\frac{1}{2}\mathbf{y}\mathbf{H}_k\mathbf{y}+\mathbf{b}_k\mathbf{y}-\delta_k$, $k=0, 1, \ldots, s$, $\mathbf{H}_k$ pozitív szemidefinit valós szimmetrikus $m\times m$-es n_k rangú mátrix, $\mathbf{b}_k\in R^m$ tetszőleges, $\delta_k\in R$ tetszőleges. Tudjuk, hogy

ekkor léteznek $\mathbf{D}_k$ teljes sorrangú mátrixok úgy, hogy $\mathbf{y}\mathbf{H}_k\mathbf{y}=(\mathbf{D}_k\mathbf{y})^T(\mathbf{D}_k\mathbf{y})$, $k=0, 1 \ldots, s$, ahol $\mathbf{D}_k: n_k \times m$. ($\mathbf{H}_k=\mathbf{D}_k^T\mathbf{D}_k$.) Alakítsuk át a feladatot:

$$\begin{array}{ll} \inf -\eta & \sup \eta \\ R_0(\mathbf{y}) \leqq -\eta & \frac{1}{2}\sum_{i=1}^{n_0}(d_i^0\mathbf{y})^2+\mathbf{b}_0\mathbf{y}+\eta-\delta_0 \leqq 0 \\ R_k(\mathbf{y}) \leqq 0, \quad k=1, \ldots, s & \frac{1}{2}\sum_{i=1}^{n_k}(d_i^k\mathbf{y})^2+\mathbf{b}_k\mathbf{y}-\delta_k \leqq 0, \quad k=1, \ldots, s. \end{array}$$

Ez egy l_p programozási primál feladat, szerkezetét a 3. ábra mutatja. (Nyilván $p_i=q_i=2$, $i\in I_k$, $k=0, 1, \ldots, s$.) A duál feladat a 3. ábra jelöléseit felhasználva:



3. ábra

$$\mathbf{x}\mathbf{A}+\mathbf{z}\mathbf{B}=-\zeta_0\mathbf{b}_0$$

$$\zeta_0=1, \quad \mathbf{z}\geqq\mathbf{0}$$

$$\zeta_k=0 \Rightarrow \xi_i=0, \quad i\in I_k, \quad k=1, \ldots, s$$

$$\inf\left\{\delta_0+\mathbf{d}\mathbf{z}+\sum_{\substack{k=0\\ \zeta_k>0}}^{s}\zeta_k\sum_{i\in I_k}\frac{1}{2}\left|\frac{\xi_i}{\zeta_k}\right|^2\right\}.$$

Ezt átalakítva:

$$\mathbf{xA}+\mathbf{zB}=-\mathbf{b}_0$$

$$\mathbf{z}\geqq\mathbf{0}$$

$$\zeta_k=0\Rightarrow\xi_i=0,\quad i\in I_k,\quad k=1,\ldots,s$$

$$\inf\left\{\delta_0+\mathbf{dz}+\frac{1}{2}\sum_{i\in I_0}(\xi_i)^2+\frac{1}{2}\sum_{\substack{k=1\\ \zeta_k>0}}^{s}\frac{1}{\zeta_k}\sum_{i\in I_k}(\xi_i)^2\right\}.$$

Megmutatjuk, hogy a lineáris feltételes kvadratikus programozási feladat esetében az l_p programozás segítségével kapott duál feladat megegyezik a szokásos duál feladattal, valamint ekkor, ha $P\neq\varnothing$ és a primál célfüggvény korlátos, akkor a duál feladatnak is van optimális megoldása.

Az l_p programozás segítségével is megállapíthatjuk, hogy amennyiben $P\neq\varnothing$ és a primál célfüggvény korlátos, akkor a duál feladatnak is van optimális megoldása, mivel a primál feladat *Slater-reguláris*, így a 4.1. tétel miatt $D^*\neq\varnothing$.

A lineáris feltételes kvadratikus programozási feladat és duálja, ahogy az VAN DE PANNE [2] könyvében is szerepel:

$$\min\left\{\frac{1}{2}\mathbf{yHy}+\mathbf{b}_0\mathbf{y}\right\}\qquad \max\left\{-\frac{1}{2}\mathbf{yHy}-\mathbf{dz}\right\}$$

$$\mathbf{By}\leqq\mathbf{d}\qquad\qquad \mathbf{yH}+\mathbf{zB}=-\mathbf{b}_0$$

$$\mathbf{z}\geqq 0.$$

Az l_p programozás segítségével nyert duál feladat:

$$\max\left\{-\mathbf{dz}-\frac{1}{2}\sum_{i=1}^{m}(\xi_i)^2\right\}$$

$$\mathbf{xD}+\mathbf{zB}=-\mathbf{b}_0$$

$$\mathbf{z}\geqq\mathbf{0}.$$

Az itt szereplő két duál feladat megegyezik, mivel a 7.3. tétel következményeiben szereplő optimalitási kritériumok miatt az $\mathbf{x}=\mathbf{yD}^T$ egyenlőségnek fenn kell állni az optimális megoldásokra és ezt az utóbbi duál feladatba behelyettesítve kapjuk az eredeti duál feladatot.

Az l_p feltételes l_p approximációs problémák.

A probléma a következő. Az F függvényt szeretnénk a legjobban l_{p_0} normában közelíteni az $f_1,\ldots,f_m$ függvények lineáris kombinációjával az $R_1,\ldots,R_u$ pontokban, feltéve, hogy a $W_1^k,\ldots,W_{t_k}^k$ pontokban a $g_1^k,\ldots,g_m^k$ függvények megfelelő lineáris kombinációja legfeljebb δ_k távolságra van a G^k függvénytől l_{p_k} normában, $k=1,\ldots,r$,

ahol $p_k > 1$, $k = 0, 1, \ldots, r$. Tehát a feladat:

$$\min \left\{ \sum_{i=1}^{u} \left| \sum_{j=1}^{m} \eta_j f_j(R_i) - F(R_i) \right|^{p_0} \right\}^{\frac{1}{p_0}}$$

$$\left\{ \sum_{i=1}^{t_k} \left| \sum_{j=1}^{m} \eta_j g_j^k(W_i^k) - G^k(W_i^k) \right|^{p_k} \right\}^{\frac{1}{p_k}} \leqq \delta_k, \quad k = 1, \ldots, r.$$

A célfüggvény minimalizálása ekvivalens η maximalizálásával, feltéve, hogy a fenti feltételeken kívül még a következő feltétel is teljesül:

$$\sum_{i=1}^{u} \frac{1}{p_0} \left| \sum_{j=1}^{m} \eta_j f_j(R_i) - F(R_i) \right|^{p_0} + \eta \leqq 0.$$

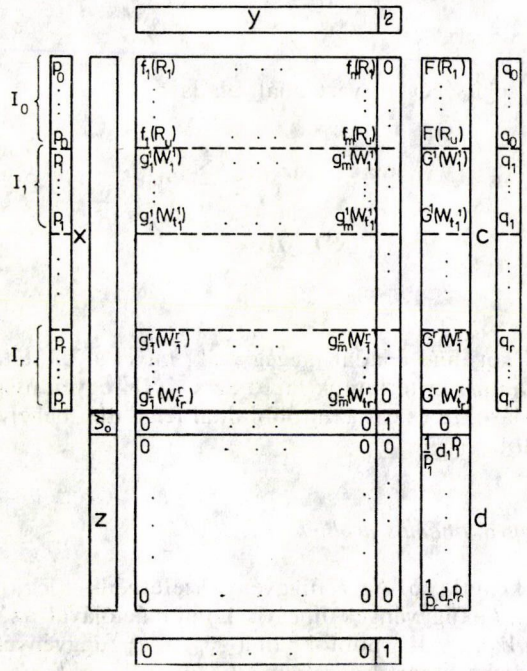
A feltételek is átalakíthatók, mindkét oldal p_k-adik hatványra emelésével és $\frac{1}{p_k}$-val való szorzással. Így a következő feladatot kapjuk:

$$\max \eta$$

$$\sum_{i=1}^{u} \frac{1}{p_0} \left| \sum_{j=1}^{m} \eta_j f_j(R_i) - F(R_i) \right|^{p_0} + \eta \leqq 0$$

$$\sum_{i=1}^{t_k} \frac{1}{p_k} \left| \sum_{j=1}^{m} \eta_j g_j^k(W_i^k) - G^k(W_i^k) \right|^{p_k} - \frac{1}{p_k} \delta_k^{p_k} \leqq 0, \quad k = 1, \ldots, r.$$

Ez a feladat pedig egy l_p programozási feladat, struktúráját a 4. ábra mutatja.



4. ábra

A duál feladat a következő a 4. ábra jelöléseit felhasználva.

$$\left(q_k = \frac{p_k}{p_k - 1},\ k = 0, \ldots, r.\right)$$

$$\mathbf{xA} = \mathbf{0}$$

$$\zeta_0 = 1$$

$$\mathbf{z} \geqq \mathbf{0}$$

$$\zeta_k = 0 \Rightarrow \xi_i = 0, \quad i \in I_k, \quad k = 1, \ldots, r$$

$$\inf\left\{\mathbf{cx} + \mathbf{dz} + \sum_{\substack{k=0 \\ \zeta_k > 0}}^{r} \frac{1}{q_k} \zeta_k \sum_{i \in I_k} \left|\frac{\xi_i}{\zeta_k}\right|^{q_k}\right\}.$$

Megjegyzés: Ez a duál feladat nyilván *Slater-regulráis.*

Kompromisszum programozás

Csak a lineáris feltételes esettel foglalkozunk. Az alapfeladat egy több célú programozási probléma. Adott egy feltételi halmaz és ezen n db célfüggvény.

$$\mathbf{By} \leqq \mathbf{d} \qquad \begin{array}{l} \min \mathbf{a}_1\mathbf{y} \\ \vdots \\ \min \mathbf{a}_k\mathbf{y} \\ \max \mathbf{a}_{k+1}\mathbf{y} \\ \vdots \\ \max \mathbf{a}_n\mathbf{y}. \end{array}$$

Tehát itt egyidejűleg több célfüggvény szerint kellene optimalizálni. Az egyes célok szerinti optimumokat jelöljük rendre $m_1, \ldots, m_k, M_{k+1}, \ldots, M_n$-nel. Így a feladat egycélúvá átfogalmazható például a következő módon.

$$\min\left\{\sum_{i=1}^{k} \alpha_i(\mathbf{a}_i\mathbf{y} - m_i)^{p_i} + \sum_{i=k+1}^{n} \alpha_i(M_i - \mathbf{a}_i\mathbf{y})^{p_i}\right\}$$

$$\mathbf{By} \leqq \mathbf{d}$$

$$p_i > 1, \quad i = 1, \ldots, n.$$

Az itt szereplő α_i, p_i, $i=1, \ldots, n$ súlyozó tényezőket a mindenkori felhasználó határozza meg. A gyakorlatban különböző súlyokkal szokás tesztelni a feladatot.

Vezessük be a következő jelöléseket:

$$\bar{\mathbf{a}}_i = \left(\frac{p_i}{\alpha_i}\right)^{\frac{1}{p_i}} \mathbf{a}_i, \quad \bar{m}_i = \left(\frac{p_i}{\alpha_i}\right)^{\frac{1}{p_i}} m_i, \quad \bar{M}_i = \left(\frac{p_i}{\alpha_i}\right)^{\frac{1}{p_i}} M_i, \quad i = 1, \ldots, n.$$

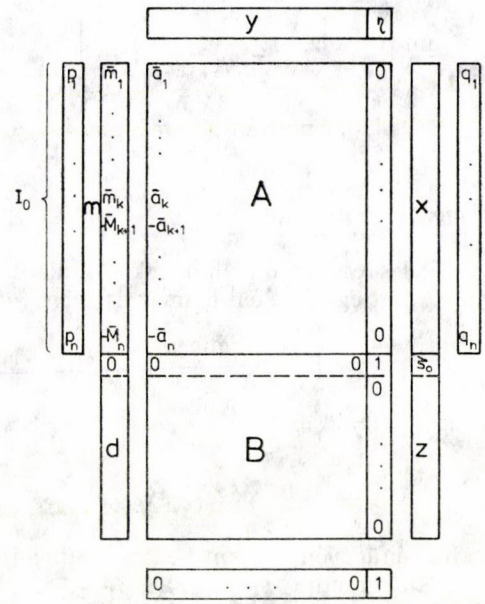
Így feladatunk a következő formában írható fel:

$$\max \eta$$

$$\sum_{i=1}^{k} \frac{1}{p_i} |\bar{\mathbf{a}}_i \mathbf{y} - \bar{m}_i|^{p_i} + \sum_{i=k+1}^{n} \frac{1}{p_i} |\bar{\mathbf{a}}_i \mathbf{y} - \bar{M}_i|^{p_i} + \eta \leqq 0$$

$$\mathbf{By} \leqq \mathbf{d}.$$

Ez pedig egy l_p programozási feladat, ahol $I_0 = \{1, \ldots, n\}$ és $I_1 = \ldots = I_k = \varnothing$. A feladat struktúráját az 5. ábra mutatja.



5. ábra

A duál feladat az 5. ábra jelöléseit felhasználva a következő:

$$\mathbf{xA} + \mathbf{zB} = \mathbf{0}$$

$$\mathbf{z} \geqq \mathbf{0}$$

$$\inf \left\{ \mathbf{mx} + \mathbf{dz} + \sum_{i=1}^{n} \frac{1}{q_i} |\xi_i|^{q_i} \right\},$$

ahol $q_i = \dfrac{p_i}{p_i - 1}$, $i = 1, \ldots, n$.

Megjegyzés: Az l_p approximációs probléma és a kompromisszum programozás esetén a feladat optimális értéke nem pozitív, mivel az $(\mathbf{x}, \mathbf{z}) = \mathbf{0}$ megoldása a duál feladatnak.

IRODALOM

[1] KLAFSZKY, E., "Geometriai programozás és néhány alkalmazása", *MTA SZTAKI Tanulmányok* 8/1973.
[2] VAN DE PANNE, C., *Methods for Linear and Quadratic Programming* (North-Holland Publishing Company, Amsterdam, Oxford, 1975).
[3] PETERSON, E. L. and ECKER, J. G., "Geometric programming: duality in quadratic programming and l_p approximation, I.", *Proceedings of the International Symposion on Mathematical Programming, ed. H. W. Kuhn and A. W. Tucker,* Princeton University Press, 1970.
[4] PETERSON, E. L. and ECKER, J. G., "Geometric programming: duality in quadratic programming and l_p approximation, II.", *SIAM Journal on Applied Mathematics* **13** (1967) 317—340.
[5] PETERSON, E. L. and ECKER, J. G., "Geometric programming: duality on quadratic programming and l_p approximation, III.", *Journal of Mathematical Analysis and Applications* **29** (1970) 365—383.
[6] STOER, J. and WITZGALL, CH., *Convexity and Optimization in Finite Dimensions, I.* (Springer-Verlag, 1970).
[7] TUCKER, A. W., "Dual systems of homogeneous linear relations", *Linear Inequalities and Related Systems, ed. H. W. Kuhn and A. W. Tucker,* Princeton University Press, 1956.

(Beérkezett: 1979. szeptember 28.)
(Újra beérkezett: 1980. március 14.)

TERLAKY TAMÁS
ELTE TTK NUMERIKUS ÉS GÉPI MATEMATIKA TANSZÉK
1088 BUDAPEST, MÚZEUM KRT. 6—8.

l_p PROGRAMMING

T. TERLAKY

Our paper treats of the primal and dual program of l_p programming. The l_p programming is the generalization of l_p approximations problems. There is strict connection between l_p programming and geometrical programming, because in both of them geometrical inequality plays a fundamental role. The structure of our paper follows that of E. KLAFSZKYS paper [1].

In the first part duality theorems are proved, which plays an important role in mathematical programming. Most of these results can be found in E. L. PETERSONS and J. G. ECKERS papers [3, 4, 5]. Afterwards the relation is investigated between the *Lagrange function* and optimal solution pair, regularity is investigated as well and we show the marginal value of l_p programming. In the end the linear programming, the l_p constrained l_p approximations problem, the quadratical constrained quadratical programming and the compromise programming are shown as special cases of l_p programming.

Alkalmazott Matematikai Lapok 6 (1980) 65—73

POLINOM-APPROXIMÁCIÓK AZ L_∞ TÉRBEN

ANDÓ GYÖRGYI és LIPCSEY ZSOLT

Budapest

A nemlineáris szűréselméletben felvetődött a polinom approximációs módszerek alkalmazása. Dolgozatunkban az (X, S, μ) véges mértékű mértéktér feletti $L_\infty(X, \mu)$ téren adunk szükséges és elegendő feltételt egy $f \in L_\infty(X, \mu)$ elemnek egy $\{\varphi_\alpha\}_{\alpha \in A}$ függvényhalmaz polinomjaival való L_∞ approximálhatóságára. Approximációs tételünk a *Stone—Weierstrass approximációs tétel* átfogalmazása mértékterekre, ahol az elválasztást a zárt intervallumok f-fel, illetve a $\{\varphi_\alpha\}_{\alpha \in A}$ elemekkel vett ősképei által generált halmazgyűrűkben fogalmazzuk meg.

1. Bevezetés

A dolgozatunkban tárgyalt approximációs probléma a szűréselmélet kapcsán vetődött fel. A szűréselmélet alapproblémáját az alábbiakban foglalhatjuk össze. Megfigyeljük az $\{y_t\}_{t=-\infty}^{\infty}$ folyamatot a $\{-\infty, t_0\}$ intervallumon. Egy X valószínűségi változóra keresünk egy adott függvényosztálybeli (pl. mérhető, folytonos, legfeljebb k-adfokú polinomok vagy lineáris) valamilyen értelemben optimális $\hat{x}(\{y_t\}_{t=-\infty}^{t_0})$ alakú becslést.

A felvetett kérdés abban tér el a numerikus approximációelméleti problémáktól, hogy az x valószínűségi változó nem áll determinisztikus kapcsolatban (legalábbis megengedett függvényosztálybeli függvénykapcsolatban) a megfigyeléseinkkel.

Ha például $\{y_t\}_{t=-\infty}^{\infty}$ tágabb értelemben stacionárius folyamat, és $x = y_{t_0+1}$, akkor a négyzetes középben legjobb lineáris szűrés feladatával csak az $\{y_t\}_{t=-\infty}^{\infty}$ folyamat lineáris regularitása esetén foglalkozunk. (Ez azt jelenti, hogy pl. egydimenziós valószínűségi változók esetében $y_{t_0+1} \notin \overline{L(\{y_t\}_{t=-\infty}^{t_0})}$, ahol L a mögötte álló függvényhalmaz lineáris burkát, a felülvonás pedig L_2-beli lezárást jelöl.) (L. pl. [6].)

A lineáris regularitásból azonban még nem következik, hogy a folyamatra vonatkozó becslési feladat a fenti értelemben szűrési probléma marad, ha a megengedett függvényosztály pl. a folytonos függvények. Nézzük példaként az alábbi folyamatot:

Legyen $\{\Omega, A, P\} = \{[0, 1], B, \lambda\}$ (B a *Borel mérhető halmazok* σ-algebrája, λ a *Lebesgue-mérték*), és $y_t(x) = e^{2\pi i t x}$, $x \in [0, 1]$. Legyen továbbá $t_0 = 0$ és $x(s) = y_1(s) = e^{2\pi i s}$. Világos, hogy a folyamat lineárisan reguláris, és a négyzetes középben legjobb lineáris becslése x-nek 0. A folytonos függvények köréből az L_∞ normában legjobb becslés létezik és

$$\hat{x}(\{e^{2\pi i t s}\}_{t=-\infty}^{t_0}) = \frac{1}{e^{-2\pi i s}} = e^{2\pi i s},$$

azaz determinisztikusan függ a megfigyeléseinktől.

A nemlineáris becslések egy másik problémája, hogy a nemlineáris becslés általában nehezen számolható, ezért közelítjük könnyen kezelhető függvényekkel, speciálisan a numerikus módszereknél már jól bevált polinomokkal. A polinomapproximációt szűrési feladatoknál is alkalmazzák (*Hermite-polinomok*, l. [7]), sőt a tágabb értelemben stacionárius folyamatok lineáris szűrése egy ilyen approximáció lineáris transzformáltja (l. [6]).

A lineáris regularitás teljesülésének eldöntésére a funkcionálanalízis sok módszert kínál. A mérhető függés ellenőrzése a gyakorlatban már sokkal nehezebb, hiszen ehhez az $\{y_t\}_{t=-\infty}^{t_0}$ függvényhalmaz által generált legszűkebb σ-algebra előállítása szükséges, és az itt szokásos approximáló függvények az általános mérhető lépcsős függvények, melyek numerikus szempontból nem a legszerencsésebbek.

Dolgozatunkban annak eldöntésére adunk használható kritériumot, hogy az x valószínűségi változó mikor approximálható L_∞ normában az $\{y(t)\}_{t=-\infty}^{t_0}$ függvényhalmaz polinomjaival. Egy további dolgozatunkban ugyanezt a problémát vizsgáljuk meg a mértékben való konvergenciára nézve. Ha az alaptér valamely K kompakt halmaz, és az $\{y_t\}_{t=-\infty}^{t_0}$ függvényosztály folytonos függvényekből áll, akkor a *Stone—Weierstrass-tétel* ad átfogó választ a kérdésre (l. pl. [9]). Az egyenletes approximáció problémáját tetszőleges X absztrakt halmaz esetében több szerző vizsgálta (l. [1], [2], [3], [4], [8]). Az approximálhatóságot az X téren az $\{y_t\}_{t=-\infty}^{t_0}$ függvényhalmaz által indukált uniform topológia segítségével jellemezték. Esetünkben ezek az eredmények azért nem alkalmazhatók, mert valószínűségi változóink, mint az L_∞ tér elemei, ekvivalencia osztályok, és így egy adott pontbeli függvényértékről általában nem beszélhetünk. A nehézséget a kommutatív *Banach-algebrák Gelfand-reprezentációja* segítségével hidaljuk át (l. [10]). Az L_∞ tér izometrikusan izomorf a maximális ideáljainak kompakt terén értelmezett folytonos függvények terével. Ezért meghatározzuk az L_∞ tér maximális ideáljait (amelyek, mint kiderül, egyértelműen megfeleltethetők az X alaptér pozitív mértékű halmazaiból álló maximális rácsoknak), és ezekre fogalmazzuk át a *Stone—Weierstrass-tétel* szükséges és elégséges feltételét.

A dolgozat tétele így pl. a [0, 1] intervallumon tartalmazza a *Stone—Weierstrass-tételt*, de annyival több, hogy jellemzi egy *Borel-mérhető* korlátos függvényhalmaz polinomjaival az $L_\infty([0, 1])$ értelemben approximálható függvényeket is.

2. Az L_∞ tér maximális ideáljai

Legyen (X, S, μ) tetszőleges véges mértékű mértéktér. Tekintsük az $L_\infty(X, S, \mu)$-t, azaz a lényegében korlátos, S-mérhető, X-en értelmezett komplex értékű függvények terét. A lényeges szuprémummal mint normával ez a tér lineáris normált tér, sőt *Banach-tér* lesz, a függvények pontonkénti szorzatával mint művelettel pedig *Banach-algebrát* kapunk (l. pl. [8]).

A későbbiekben hasznunkra lesz és önmagában is érdekes az $L_\infty(X, S, \mu)$ függvényalgebra szerkezetébe mélyebb betekintést nyújtó következő tétel az algebra maximális ideáljairól.

Az $L_\infty(X, S, \mu)$ függvényalgebrát röviden L_∞-nel fogjuk jelölni: ez tehát egységelemes kommutatív *Banach-algebra* és mint ilyen biztosan tartalmaz maximális valódi ideálokat. Ezek olyan zárt L_∞-beli halmazok, amelyek csupa L_∞-ben nem

invertálható elemből állnak, azaz olyan függvényekből, amelyeknek lényeges értéke a 0. (Egy $f \in L_\infty$ függvény lényeges értéke az a komplex szám, ha a tetszőleges komplex síkbeli környezetének f szerinti ősképe X-ben pozitív mértékű halmaz.)

Tekintsünk most egy $f \in L_\infty$ függvényt, amelyik L_∞-ben nem invertálható. Ez utóbbi tulajdonságból és a lényeges érték definíciójából következik, hogy a C komplex számsíkon a 0 körüli ε sugarú környezetek f szerinti ősképe X-ben pozitív mértékű halmazokból álló halmazrendszer lesz, amelyről könnyen belátható, hogy rácsot alkot X-ben. (Egy halmazrendszert akkor nevezünk rácsnak, ha nem üres, nem tartalmazza az üres halmazt és bármely két a rendszerhez tartozó halmaz metszete tartalmaz valamely a rendszerhez tartozó halmazt.) Az L_∞-beli maximális ideálok és bizonyos X-beli maximális rácsok kapcsolatáról szól a következő tétel.

2.1. TÉTEL. Az L_∞ algebra maximális ideáljai és az X-beli pozitív mértékű halmazokból álló maximális rácsok kölcsönösen egyértelműen megfeleltethetők egymásnak.

Bizonyítás. Legyen H maximális ideál az L_∞-ben, és tekintsük az összes $f \in H$ függvényre és minden $\varepsilon > 0$-ra a $0 \in C$ ε sugarú környezetének f szerinti ősképét X-ben. Mivel a 0 minden $f \in H$ függvénynek lényeges értéke, az ősképek mind pozitív mértékű halmazok. Tetszőleges két őskép metszete szintén pozitív mértékű halmaz, ugyanis ha 0-mértékű, vagy üres lenne valamely $A_f \cap A_g$, ahol A_f, illetve A_g a 0 egy környezetének f, ill. g szerinti ősképe, akkor az $|f|^2+|g|^2 \in H$ függvény invertálható lenne, ami ellentmondás. Jelöljük φ_ε-nal a $0 \in C$ ε-sugarú környezetének az $f \in H$ függvények szerinti ősképeit. Az előzőek szerint φ_ε pozitív mértékű halmazokból álló rács X-en.

Legyen Φ az X pozitív mértékű halmazaiból álló rácsok összessége. Φ-n a következő részben rendezést adjuk meg: a $\varphi^1 \in \Phi$ rács finomabb, mint $\varphi^2 \in \Phi$ (jelölésben $\varphi^1 \succ \varphi^2$), ha tetszőleges $A \in \varphi^2$-höz létezik $B \in \varphi^1$, hogy $B \subset A$. Mivel Φ minden teljesen rendezett részhalmazának van felső korlátja Φ-ben — mégpedig a részhalmazbeli rácsok egyesítése, ami szintén rács — a *Zorn-lemma* biztosítja, hogy minden $\varphi \in \Phi$ rácshoz van őt tartalmazó t maximális rács Φ-ben. $\varepsilon_1 < \varepsilon_2$ esetén $\varphi_{\varepsilon_1} \succ \varphi_{\varepsilon_2}$, így a φ_ε rácsok összessége teljesen rendezett halmazosztályt alkot és az előzőek szerint Φ-ben van a φ_ε rácsokat tartalmazó maximális elem.

Egy maximális ideálhoz csak egy $t \in \Phi$ maximális rács tartozhat. Ugyanis, ha $t_1 \neq t_2$ Φ-beli maximális rácsok, akkor tetszőleges $A \in t_1$ halmazhoz létezik olyan $B \in t_2$ halmaz, hogy $A \cap B = \varnothing$ vagy $\mu(A \cap B) = 0$. Mivel feltevésünk szerint t_1 és t_2 a H maximális ideálhoz tartozó maximális rácsok, létezik olyan f és $g \in H$ függvény, hogy a $0 \in C$ valamilyen ε sugarú ősképének f, illetve g szerinti ősképe A-ba, illetve B-be esik. Ekkor azonban az $|f|^2+|g|^2 \in H$ függvény invertálható lenne L_∞-ben, ami ellentmondás.

Másrészt nyilvánvaló, hogy ha $t \in \Phi$ maximális rács (ami tehát csupa pozitív mértékű halmazból áll), akkor azok az $f \in L_\infty$ függvények, amelyekre a $0 \in C$ körüli ε sugarú ősképek t-ben vannak, egy $H \subset L_\infty$ maximális ideált alkotnak.

3. Approximálhatóság az L_∞ térben

Előzetes megjegyzések.

Az L_∞-beli függvények pozitív mértékű nívóhalmazai és az X pozitív mértékű halmazaiból álló maximális rácsok központi szerepet játszanak a bevezetésben megfogalmazott probléma megoldásában. Előzetesként ezekkel kapcsolatban hivatkozunk néhány tényre.

Jelöljük T-vel az X mérhető halmazaiból álló maximális rácsok terét. Tetszőleges $A \in S$ halmazhoz rendeljünk hozzá egy T-beli halmazt a következőképpen: $\gamma(A) = \{t, t \in T, A \in t\}$. A γ leképezés X-beli mérhető halmazokhoz T-beli halmazokat rendel, kölcsönösen egyértelmű és *Boole-művelettartó* leképezés, így az S σ-algebra képeiként kapott halmazok S' algebrát alkotnak T-ben. A *Boole-algebrák* reprezentációjára vonatkozó *Stone-tétel* szerint ezen a T halmazon létezik egy teljesen szétesső, lokálisan kompakt topológia, amelyet az S' mint bázis segítségével adhatunk meg, és amelyben az S'-beli halmazok nyíltak és kompaktak is (l. [11], [12], valamint [5]). A T tér bizonyos részhalmazán értelmezett folytonos függvényekre alkalmazott *Stone—Weierstrass-tétel* segítségével fogunk következtetéseket levonni az L_∞-beli függvényekre vonatkozóan.

Legyen $f \in L_\infty$ tetszőleges függvény és legyenek $\{g_\alpha\}_{\alpha \in A}$ szintén tetszőleges L_∞-beli függvények, ahol α valamilyen indexhalmaz. Jelöljük $\overline{\{P(g)_\alpha\}}$-val a $\{g_\alpha\}$ függvényrendszer polinomjainak L_∞ lezárását és $R\{(g)_\alpha\}$-val a $\{g_\alpha\}$ függvények olyan pozitív mértékű nívóhalmazai által generált gyűrűt, amelyek a komplex sík zárt és diszjunkt halmazainak a g_α függvények szerinti ősképeiként állnak elő.

Jelöljük $g\{I^k\}$-val a k-dimenziós zárt intervallumok által generált gyűrűt.

3.1. DEFINÍCIÓ. Azt mondjuk, hogy a $\{g_\alpha\}$ függvényrendszer elválasztja az X tér $\mu(A \cap B) = 0$ tulajdonsággal rendelkező pozitív mértékű A, ill. B halmazát, ha A-hoz és B-hez létezik olyan k természetes szám, G_1, és $G_2 \in g\{I^k\}$ halmazok, amelyekre $G_1 \cap G_2 = 0$, valamint olyan $(g_{\alpha_1}, \ldots, g_{\alpha_k}) \in \{g_\alpha\}$ függvényrendszer, hogy a $g_k : x \to (g_{\alpha_1}(x), \ldots, g_{\alpha_k}(x)) \in R^k$ leképezésre $g_k^{-1}(G_1) \supset A$ és $g_k^{-1}(G_2) \supset B$.

A fő tétel:

3.1. TÉTEL. Az f függvény akkor és csak akkor eleme $\overline{\{P(g)_\alpha\}}$-nak, ha az f által elválasztott X-beli pozitív mértékű halmazokat a $\{g_\alpha\}$ függvényrendszer is elválasztja.

Bizonyítás. Jelöljük K-val az X-beli csupa pozitív mértékű halmazból álló maximális rácsok γ képeinek halmazát T-ben. Ha $f \in L_\infty$, akkor $\gamma(f^{-1}(\sigma(f))) \subset K$, ahol $\sigma(f)$ az f lényeges értékeinek halmaza; $\sigma(f)$ kompakt, nem üres halmaz a komplex számsíkon.

3.1. LEMMA. Ha $t \in K$, akkor létezik éspedig egyetlen olyan λ lényeges értéke f-nek, hogy t éppen az adott λ-hoz tartozó maximális rács (a 2.1. tételből következik, hogy tetszőleges $\lambda \in \sigma(f)$ lényeges értékhez X-ben csupa pozitív mértékű halmazból álló maximális rácsok tartoznak).

Bizonyítás. Legyen $t \in K$ maximális rács. Vegyük most a $\sigma(f) \subset C$ halmaz egy lefedését $1/n$ sugarú körökkel. Ezek f szerinti őspei 0 mértékű halmaztól eltekintve X egy pozitív mértékű halmazokkal való lefedését adják. Ha $t \in K$ tetszőleges maximális rács, akkor létezik olyan B_n halmaz a lefedő rendszerből, hogy minden $A \in t$ halmazzal vett metszete pozitív mértékű, mert ha ilyen B_n nem létezne, akkor a lefedő rendszerhez X-ben hozzáadhatnánk tőle diszjunkt pozitív mértékű halmazt, ami ellentmondás. Ebből következik, hogy $B_n \in t$. A felosztás finomításával B_n-ben egymásba skatulyázott halmazsorozatokat kapunk, amelyek között mindig lesz t-beli elem (ez az előbbi gondolatmenet ismétlésével azonnal adódik).

Mivel t tetszőleges volt és a t maximális rácsban nem lehetnek diszjunkt halmazok, nyilvánvaló, hogy f-nek létezik, éspedig egyetlen olyan lényeges értéke, amelyhez a t maximális rács tartozik.

Tehát $K \subset \gamma(f^{-1}(\sigma(f)))$. Mivel pedig X-beli mérhető halmazok γ szerinti képe T-ben nyílt és kompakt halmaz, K kompakt részhalmaza T-nek.

Tekintsük K-n a következő függvényt: $t \in K$-ra $f'(t) = \lambda$, ahol λ az $f \in L_\infty$ függvény t maximális rácshoz tartozó lényeges értéke. Nyilvánvaló, hogy az így értelmezett f' függvény folytonos K-n.

Nézzük tehát az $f'(t) \in C(K)$ függvényt és a $\{g'_\alpha\}$ függvényeket, amelyek szintén K-n értelmezett folytonos függvények. Könnyen látható, hogy az f' és $g' \in C(K)$ függvények normája megegyezik a nekik megfelelő f, illetve $g \in L_\infty$ függvények L_∞ normájával.

A *Stone—Weierstrass-tételt* alkalmazva ezekre a függvényekre, azt kapjuk, hogy f' akkor és csak akkor eleme $\overline{P\{g'_\alpha\}}$-nak, ha azokat a $t \in K$ pontokat, amelyeket f' elválaszt, $\{g'_\alpha\}$ is elválasztja, azaz ha $t_1 \neq t_2$ esetén $f'(t_1) \neq f'(t_2)$, akkor létezik olyan $\alpha \in \mathsf{A}$, hogy $g'_\alpha(t_1) \neq g'_\alpha(t_2)$.

Tegyük fel most, hogy $f \in \overline{\{P(g)_\alpha\}}$. Legyenek $t_1 \neq t_2 \in K$ olyan maximális rácsok, amelyeket f' és ezért $\{g'_\alpha\}$ is elválaszt, azaz létezik olyan λ_1, λ_2, hogy $\lambda_1, \lambda_2 \in \sigma(f)$, $f'(t_1) = \lambda_1$, $f'(t_2) = \lambda_2$ és $\lambda_1 \neq \lambda_2$; valamint valamilyen $\varepsilon < \frac{|\lambda_1 - \lambda_2|}{2}$-re az $A = \{x: |f(x) - \lambda_1| \leqq \varepsilon\}$ és $B = \{x; |f(x) - \lambda_2| \leqq \varepsilon\}$ jelölés mellett $A \in t_1$, $B \in t_2$ és $\mu(A \cap B) = 0$.

3.2. LEMMA. Ha $\{g'_\alpha\}$ elválasztja K-n az f' által elválasztott maximális rácsokat, akkor $R\{(g)_\alpha\}$ elválasztja a fenti módon megadott tetszőleges A, B párokat.

Bizonyítás. Először is elkészítjük az A, illetve B halmaz egy-egy lefedését, azután ezeket alkalmas módon úgy választjuk meg, hogy metszetük mértéke 0 legyen. f' tehát elválasztja t_1-et és t_2-t, és ha $\{g'_\alpha\}$ is elválasztja, ez azt jelenti, hogy létezik olyan $\alpha \in \mathsf{A}$, hogy $g'_\alpha(t_1) \neq g'_\alpha(t_2)$, azaz van olyan $v^1_\alpha \neq v^2_\alpha$ lényeges értéke g_α-nak, hogy ezek valamilyen környezetének g_α szerinti őspei t_1-ben, illetve t_2-ben vannak, vagyis $\mu\big(g_\alpha^{-1}(G(v^1_\alpha, \varepsilon)) \cap A\big) > 0$ és $\mu\big(g_\alpha^{-1}((v^2_\alpha, \varepsilon)) \cap B\big) > 0$ valamilyen $\varepsilon > 0$-ra. f' azonban elválasztja mindazokat a maximális rácsokat, amelyekhez az A, illetve a B halmaz hozzátartozik, és ugyanezt megteszi a $\{g'_\alpha\}$ függvénycsalád is, tehát jelöljük v^A-val azon v^1_α lényeges értékek halmazát, amelyekre igaz, hogy valamilyen $\alpha \in \mathsf{A}$-ra és $\varepsilon > 0$-ra $g_\alpha^{-1}(G(v^1_\alpha, \varepsilon)) \in t$, ahol $t \in \gamma(A) \subset K$, azaz $\mu\big(g_\alpha^{-1}(G(v'_\alpha, \varepsilon)) \cap A\big) > 0$. Ugyanígy elkészítve a v^B halmazt, a $g_\alpha^{-1}(G(v^1_\alpha, \varepsilon))$, illetve $g_\alpha^{-1}(G(v^2_\alpha, \varepsilon))$ halmazokkal mindenesetre az A, illetve a B halmaz egy-egy lefedését kapjuk.

Legyen most $t_1 \in \gamma(A)$ tetszőleges. Akkor akárhogyan is válasszuk meg a $t \in \gamma(B)$ maximális rácsot, van olyan $\alpha \in \mathsf{A}$ és $v_\alpha^1 \in v^A$, illetve $v_\alpha^2 \in v^B$, hogy

$$g_\alpha^{-1}(G(v_\alpha^1, \varepsilon)) \in t_1, \quad g_\alpha^{-1}(G(v_\alpha^2, \varepsilon)) \in t \quad \text{és} \quad \gamma(g_\alpha^{-1}(G(v_\alpha^1, \varepsilon))) \cap \gamma(g_\alpha^{-1}(G(v_\alpha^2, \varepsilon))) = \varnothing$$

valamilyen alkalmas $\varepsilon > 0$-ra. Így a rögzített $t_1 \in \gamma(A)$ ponthoz a $\gamma(B)$ halmaz egy lefedését kaptuk, és mivel $\gamma(B)$ kompakt halmaz, valamilyen n-re $\gamma(B) \subset \subset \bigcup_{i=1}^{n} \gamma(g_{\alpha_i}^{-1}(G(v_{\alpha_i}^2, \varepsilon)))$. Ekkor viszont a $\bigcap_{i=1}^{n} \gamma(g_{\alpha_i}^{-1}(G(v_{\alpha_i}^1, \varepsilon)))$ halmaz a $t_1 \in \gamma(A)$ pontnak a $\gamma(B)$ előző lefedésétől diszjunkt környezete. $\gamma(A)$ ilyen módon való lefedéséből szintén kiválasztható véges sok, ami még mindig lefedi $\gamma(A)$-t, azaz létezik olyan m, hogy $\bigcup_{j=1}^{m} \bigcap_{i=1}^{n} \gamma(g_{\alpha_{ij}}^{-1}(G(v_{ij}^2, \varepsilon))) \supset \gamma(A)$. De akkor $\bigcap_{j=1}^{m} \bigcup_{i=1}^{n} \gamma(g_{\alpha_{ij}}^{-1}(G(v_{ij}^2, \varepsilon))) \supset \gamma(B)$ és a két lefedő halmazrendszer diszjunkt lesz.

A 3.1. tétel fordított irányú állítása rögtön következik a 3.3. lemmából.

3.3. LEMMA. Ha $\{g_\alpha\}$ elválasztja az előzőekben megadott tetszőleges A, B halmazpárt, akkor $t_1, t_2 \in K$, $t_1 \neq t_2$, $f'(t_1) \neq f'(t_2)$ esetén létezik olyan $\alpha \in \mathsf{A}$, hogy $g_\alpha'(t_1) = g_\alpha'(t_2)$.

Bizonyítás. Feltevéseink szerint tetszőleges $t_1 \in \gamma(A)$ és $t_2 \in (B)$ párra fennáll, hogy $f'(t_1) \neq f'(t_2)$. Mivel azonban $\{g_\alpha\}$ elválasztja A-t és B-t, az elválasztás definíciójának alkalmazásával a lemma állítása azonnal adódik.

4. Egy alkalmazás

A 3. pontban bebizonyított tétel egy kézenfekvő elméleti jellegű alkalmazásaként megmutatjuk, hogy a szeparábilis mértékterek klasszikus reprezentációs tételét (P. R. HALMOS, J. v. NEUMAN, V. A. ROCHLIN, [9]) igen egyszerűen megkaphatjuk.

Legyenek most $f_1(x), f_2(x), \ldots, f_n(x), \ldots$ L_∞-beli függvények. Akkor megadható olyan $F(x) \in L_\infty$ függvény, hogy az $\{f(x)\}_{i=1}^{\infty}$ függvények által generált algebra L_∞ lezárása megegyezik az $F(x)$ függvény polinomjainak L_∞ lezárásával kapott függvényhalmazzal, vagyis az $\{f_i(x)\}_{i=1}^{\infty}$ függvények és $F(x)$ ugyanazokat a mérhető függvényeket állítják elő. Az $F(x)$ megadása a következő módon történhet.

Legyen $f_1(x)$ és $f_2(x)$ két L_∞-beli nemnegatív függvény. Tekintsük az $f_1(x)$-hez tartó lépcsős függvények sorozatából az n-edik függvényt és legyenek az A_i^n, $i = 0, \ldots, 2^n$ halmazok az $f_1^n(x)$ nívóhalmazai, azaz

$$A_i^n = \left\{x;\ x \in X,\ f_1(x) \in \left[\frac{i}{2^n} n, \frac{i+1}{2^n} n\right)\right\}, \quad i = 0, 1, \ldots, 2^n - 1$$

és

$$A_{2^n}^n = \{x;\ x \in X,\ f_1(x) \geqq n\}.$$

Ugyanígy az $f_2^n(x)$ nívóhalmazai B_i^n, $i = 0, 1, \ldots, 2^n$. Vezessük be a $D_{ij}^n = A_i^n \cap B_j^n$, $i, j = 0, 1, \ldots, 2^n$ jelölést. Akkor $\bigcup_{i,j=1}^{2^n} D_{ij} = X$, $D_{ij}^n \cap D_{lk}^n = \varnothing$, ha $(i, j) \neq (l, k)$, és az

A_i^n halmaz előáll $A_i^n = \bigcup_{j=0}^{2^n} D_{ij}^n$ alakban. Legyen $s = \frac{n}{2^n}/3^{n+1}$, akkor készítsük el a következő lépcsős függvényt.

$$F^1(x) = \sum_{i=0}^{2^n} \sum_{l=0}^{2^n} \left(\frac{n}{2^n} i + ls\right) \chi_{il}^n = \sum_{k=0}^{L} a_k E_k,$$

ahol az E_k halmazok rendre megegyeznek a $D_{00}, D_{01}, \ldots, D_{02^n}, D_{10}, \ldots, D_{12^n}, \ldots$ halmazokkal, az a_k számok pedig ugyanilyen módon sorba rendezve az $F^1(x)$-beli együtthatókat jelentik, az a_k, $k=0, \ldots, L$ sorozat tehát szigorúan monoton növő számsorozat.

Az $F^2(x)$ függvényt úgy kapjuk, hogy az F^1 nívóhalmazait sorra elmetsszük az $f_1^{n+1}, f_2^{n+1}, f_3^{n+1}$ nívóhalmazaival és az újonnan keletkezett halmazokhoz az előbb leírt módon rendelünk értékeket. Ha az F^1 elkészítésénél $n=2$-t veszünk, akkor világos, hogy $F^n(x)$ minden egyes nívóhalmaza 2^{2n+2} diszjunkt részre bomlik az $F^{n+1}(x)$ előállításakor, tehát ha

$$F^n(x) = \sum_{k=0}^{L} a_k E_k$$

alakú volt, akkor

$$F^{n+1}(x) = \sum_{k=0}^{L} \sum_{l=0}^{2^{2n+2}-1} (a_k + b_k \cdot l) \chi D_{kl},$$

ahol $b_k = (a_{k+1} - a_k)/3^{2n+2}$.

Ezen a módon egy $F^1, F^2, \ldots, F^n, \ldots$ lépcsős függvénysorozatot kapunk, mely ről könnyen látható, hogy pontonként konvergens, azaz egy $F(x)$ mérhető függvényhez tart, valamint ha az $\{f_i\}_{i=1}^\infty$ függvényrendszer elválaszt bizonyos pozitív mértékű halmazokat, akkor $F(x)$ is megteszi ugyanezt. Az $F(x)$ függvény még a következő tulajdonságokkal is rendelkezik:

1. A 2. pont szerint minden $f_i(x)$, $i=1, 2, \ldots$ függvényhez megadható az $F \in L_\infty$, $\sigma(F) \subset R$ kompakt halmazon olyan φ_i folytonos függvény, hogy $f_i = \varphi_i \circ F$.

2. F mérhető az $\{f_i\}_{i=1}^\infty$ függvényrendszer által generált legszűkebb σ-algebrára, valamint az f_i, $i=1, 2, \ldots$ függvények mérhetők az F által generált legszűkebb σ-algebrára, vagyis ez a két σ-algebra egybeesik.

3. A $\sigma(F)$ halmaz $B(\sigma(F))$ *Borel-halmazain* megadható egy $\hat{\mu}$ valószínűségi mérték a

$$\hat{\mu}(B) = \mu(F^{-1}(B)), \quad B \in B(\sigma(F))$$

definícióval (feltéve, hogy $\mu(X)=1$ volt), ahol még az is fennáll, hogy a $\varphi: B \to F^{-1}(B)$, $B \in B(\sigma(F))$ leképezés izomorfizmus az F által generált legszűkebb σ-algebra és $B(\sigma(F))$ között. Mindezeket figyelembe véve tehát igaz az alábbi

4.1. TÉTEL. Tetszőleges megszámlálható $\{f_i\}_{i=1}^\infty \in L_\infty(X, S, \mu)$, $(\mu(X)=1)$ függvényrendszerhez megadható egy $K \subset R$ kompakt halmaz, egy

$$\varphi: \sigma(\{f_i\}_{i=1}^\infty) \to B(K)$$

σ-izomorfizmus (itt $\sigma(\{f_i\}_{i=1}^{\infty})$ az f_i, $i=1, 2, \ldots$ függvények által generált legszűkebb σ-algebrát jelenti, $B(K)$ pedig a K kompakt halmaz *Borel-halmazait*), valamint egy $\hat{\mu}$ valószínűségi mérték, úgyhogy a

$$\varphi\colon (X, \sigma(\{f_i\}_{i=1}^{\infty}), \mu) \to \{K, B(K), \hat{\mu}\}$$

leképezés mértéktartó σ-izomorfizmus. $C(\{f_i\}_{i=1}^{\infty})$-vel jelölve az $\{f_i\}_{i=1}^{\infty}$ függvények polinomjainak L_∞-lezártját és $C(K)$-val a K-n értelmezett folytonos függvényeket, a $C(\{f_i\}_{i=1}^{\infty})$ és a $C(K)$ *Banach-algebrák* izomorfak egymással.

4.2. TÉTEL. A 4.1. tétel feltételei mellett megadható egy

$$\varphi_\sigma\colon B([0, 1]) \to \sigma(\{f_i\}_{i=1}^{\infty})$$

σ-homomorfizmus és egy $0 \leqq \alpha \leqq 1$ szám, úgyhogy $B \in B([0, \alpha]) \subset B([0, 1])$ esetén

$$\lambda(B) = \mu(\varphi_\sigma(B)), \quad \text{azaz} \quad \hat{\mu} = \lambda.$$

(λ a *Lebesgue-mértéket* jelöli). $B \in B([\alpha, 1])$ esetén pedig $\hat{\mu}$ legfeljebb megszámlálható sok $1-\alpha$ összmértékű egypontmérték összege. Ha $(X, \sigma(\{f_i\}_{i=1}^{\infty}), \mu)$ nem atomos, akkor $\alpha=1$. A $\varphi_c\colon C(\{f_i\}_{i=1}^{\infty}) \to L_\infty([0, 1], B, \hat{\mu})$ leképezés *Banach-algebra izomorfizmus*.

Szeparábilis mértékteret generáló tetszőleges megszámlálható függvényrendszerhez mindezek szerint vagy olyan reprezentáció adható meg, ahol a függvényrendszer elemei folytonosak (4.1. tétel), vagy olyan reprezentáció, amelyben — nem atomos szerkezetű mértékteret feltételezve — $\hat{\mu}$ éppen a λ *Lebesgue-mérték* lesz (4.2. tétel). Mindkét reprezentáció megadható abban az esetben, ha F előállítható olyan módon, hogy $\sigma(F) \subset R$ spektruma összefüggő halmaz.

A 4.2. tétel bizonyításával kapcsolatban megjegyezzük, hogy egyszerű transzformációk eredménye. Legyen ugyanis $\hat{H}(y) = \mu\{x;\ F(x) \leqq y\}$. Mivel $0 \leqq \hat{H}(y) \leqq 1$ és $\hat{H}(y)$ monoton növő függvény, előállítható a $\hat{H}(y) = H_f(y) + H_s(y)$ alakban, ahol $H_f(y)$ szigorúan monoton növő folytonos függvény, $H_s(y)$ pedig tiszta ugrófüggvény. Legyen $\alpha = \sup_y H_f(y)$ és készítsük el a

$$H(y) = \begin{cases} H_f(y), & \text{ha } y \ \hat{H}(y) \text{ folytonossági pontja,} \\ \alpha + H_s(y), & \text{ha } y \text{ nem folytonossági pontja } \hat{H}(y)\text{-nak} \end{cases}$$

függvényt. Könnyen látható, hogy az $\hat{E} = H \circ F$ függvény teljesíti a σ-algebrákra vonatkozó 2. feltételt, így a 4.2. tétel állításai azonnal következnek.

A szűréselmélet szempontjából fontos, mértékben, illetve L_p-normában való approximációt egy következő dolgozatunkban fogjuk megvizsgálni. Ugyanitt térünk majd ki a feltételes várható érték approximációs problémáinak vizsgálatára is.

IRODALOM

[1] CSÁSZÁR, Á., CZIPSZER, J., „Sur des critères généraux d'approximation uniforme", *Annales Univ. Scient. Budapestinensis de R. Eötvös nom. Sect. Math.* **VI.** (1963).

[2] CSÁSZÁR, Á., „Sur un critère d'approximation uniforme", *Publications of the Mathematical Inst. of the Hungarian Acad. of Sciences*, **VIII.** Ser. B, 1964.

[3] CSÁSZÁR, Á., „Gleichmässige Approximation und Gleichmässige Stetigkeit", *Acta Mathematica Scientiarum Hungaricae* **20** (1969) 253—261.

[4] CSÁSZÁR, Á., "On approximation theorems for uniform spaces", *Acta Mathematica Acad. Sci. Hun.* **22** (1971) 177—186.
[5] DINCULEANU, N., *Vector Measures* (Veb. Deutscher Verlag der Wissenschaften, Berlin, 1966).
[6] DOOB, J. L., *Stochastic Processes* (John Wiley, New York, 1953).
[7] IBRAHIMOV, I. A., ROZANOV, J. A., *Gauszovszkie szlucsajnüe processzü* (Izd. Nauka, Moszkva, 1970).
[8] NÖBELING, G., BAUER, H., „Allgemeine Approximations Kriterion mit Anwendungen", *Jber. Deutsch. Math.* **58** (1955) 54—72.
[9] ROCHLIN, V. A., „Ob osznovnüh ponyatijah teorii merü", *Mat. Szbornik* **25/67** (1949) 107—150.
[10] RUDIN, W., *Functional Analysis* (McGraw-Hill, New York, 1973).
[11] STONE, N. H., "The theory of representation of *Boolean algebras*", *Trans, Amer. Math. Soc.* **40** (1936) 37—111.
[12] STONE, N. H., "Applications of the theory of *Boolean rings* to general topology", *Trans. Amer. Math. Soc.* **41** (1949) 375—481.

(Beérkezett: 1980. január 21.)

ANDÓ GYÖRGYI ÉS LIPCSEY ZSOLT
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1132 BUDAPEST, VICTOR HUGO U. 18.

POLYNOM APPROXIMATIONS IN SPACE L_∞

GY. ANDÓ and ZS. LIPCSEY

In the nonlinear filtering theory the necessity to use the approximation by polinomials arises. In the present paper we consider the problem of approximation by polynomials in $L_\infty(X, \mu)$ with a finite measure space (X, S, μ). *Stone's and Weierstrass's approximation theorem* is formulated by means of the rings generated by the preimages of the closed intervals with on one hand the approximating functions. This results in a necessary and sufficient condition for the approximability problem.

Alkalmazott Matematikai Lapok 6 (1980) 75—82

NUMERIKUS MÓDSZER KONVEX FÜGGVÉNY LEGJOBB MEGKÖZELÍTÉSÉRE, E FÜGGVÉNY *N* SZÁMÚ, ÁLTALUNK MEGHATÁROZHATÓ PONTOKBAN FELVETT ÉRTÉKEI ALAPJÁN

FEUER GÁBOR

Budapest

A cikk egy olyan új numerikus módszert ismertet, amelynek célja konvex függvények legjobb közelítése, feltéve, hogy az adott függvény értékeit *N* számú általunk meghatározható pontban ismerjük. A módszer a két végpontban feltételezi a deriváltak ismeretét is. Mindezek ismeretében célunk, hogy a konvex függvénynek (egy adott intervallumot nézve) az ismeretlen függvényértékű pontjaira vonatkozó bizonytalanságot minimalizáljuk és ilyen feltétel mellett határozzunk meg *n* olyan újabb mérési helyet, amelyhez tartozó függvényértékek (mért értékek) segítségével a lehető legnagyobb mértékben csökkentjük a konvex függvény alakjára vonatkozó bizonytalanságot.

1. Bevezetés

A most ismertetésre kerülő numerikus módszer függvények közelítésére vonatkozik, abban az esetben, ha a szóban forgó függvényt, illetve ennek deriváltját két adott pontban ismerjük. A két adott pont közötti intervallumban bárhol mérést, illetve méréseket végezhetünk a függvény értékére vonatkozóan. Célunk az, hogy adott, *N* számú mérés optimális elhelyezésével minél pontosabb információt szerezzünk egy konvex függvény alakjára vonatkozóan.

Az „optimális approximáció" ilyen jellegű felvetése a szovjet, az amerikai és a magyar matematikai szakirodalomban egyaránt ismeretes. A szovjet szakirodalmat nézve legjelentősebb N. Sz. BAHVALOV *„A gépi matematika numerikus módszerei"* című könyve, l. [1], az amerikai folyóiratokban pl. C. A. MICHELLI foglalkozott hasonló problémákkal, l. [2]. Nálunk Magyarországon nemrég készült el e témával kapcsolatban SONNEVEND GYÖRGY *„On the optimization of adaptive algorithms"* című munkája, l. [3].

A cikk első részében a módszer elve kerül ismertetésre $n=1$ esetben, a második rész olyan általánosítást tartalmaz, amely tetszőleges n-re alkalmazható.

Tekintsük először azt az esetet, amikor a két adott pont közötti intervallumban egy mérési helyet kell optimálisan meghatározni. A konvexitás feltételéből következik, hogy a függvény gráfja a két végpontot és a mért értéket összekötő törtvonal fölött nem helyezkedhet el. Ehhez hasonlóan belátható, hogy a függvény nem vehet fel értéket sem az érintők, sem pedig a mért függvényértékekhez húzott törtvonal mindkét oldali meghosszabbítása alatt. Így bizonyosak lehetünk abban, hogy a szóban forgó konvex függvény a mért érték mindkét oldalán az előbb meghatározott *ABP* és *CDP* (lásd: 1-2. ábra) háromszögek területén belül veszi fel értékét. Ezek után célszerű a mérési helyet úgy megválasztani, hogy a keletkező háromszögek a lehető legkisebb bizonytalanságot tartalmazzák a függvény alakjára vonatkozóan. Ehhez azonban mérnünk kell ezt a bizonytalanságot. Legyen ez a mérőszám a

függvényértékek maximálisan lehetséges etérése. Jelöljük a jobb oldali, illetve bal oldali maximális lehetséges eltérést az optimális mérési hely függvényében $\xi_1(\tau)$-val, illetve $\xi_2(\tau)$-val. A legkisebb bizonytalanságot a konvex függvény alakjára vonatkozóan olyan mérési hely mellett kapjuk, amelyre $\xi_1(\tau)=\xi_2(\tau)=\inf\max(\xi_1, \xi_2)$.

1.1. LEMMA. Optimális megoldás $n=1$ esetben annál a mérési helynél lesz, amelynél a két maximális lehetséges eltérés egyenlő.

Bizonyítás: Ha a mért értéket rögzítjük, akkor mindkét oldalon a maximális lehetséges eltérés a mérési hely függvényében folytonosan változik. A mérési helyet balról jobbra változtatva a jobb oldali maximális lehetséges eltérés monoton csökken, míg a bal oldali maximális lehetséges eltérés monoton nő. Jelöljük a bal oldali maximális lehetséges eltérés és a mérési hely összefüggését leíró függvényt $\{\xi_1$-gyel, a jobb oldaliét pedig ξ_2-vel. Keressük meg most az optimális mérési hely feltételének megfelelő $\inf\max(\xi_1, \xi_2)$ pontot. Mivel ξ_1 és ξ_2 egyaránt folytonos az adott [0, 1] intervallumon, ami egyben azt is jelenti, hogy $0\leqq\xi_1\leqq M_1$ és $0\leqq\xi_2\leqq M_2$ mellett létezik olyan mérési hely az adott intervallumon, amelyre a $\max(\xi_1, \xi_2)$ függvénynek minimuma van. A minimum unicitását ξ_1 monoton csökkenő és ξ_2 monoton növekvő volta biztosítja.

Tehát létezik egy olyan τ optimális mérési hely, amelyre $\xi_1(\tau)=\xi_2(\tau)=$ $=\inf\max(\xi_1, \xi_2)$. Optimális megoldás annál a mérési helynél lesz, ahol a két maximális lehetséges eltérés egyenlő. Ez lényegesen egyszerűsíti a problémát, hiszen ezek után nem kell mást tenni, mint mindkét oldalon megadni a maximális lehetséges eltéréseket a mért érték függvényében, az így kapott függvények maximumát kiszámítani, majd behelyettesítés után az így már csak a mérési helytől függő két függvényt egyenlővé tenni. Az egyenlet megoldása által $n=1$ esetre optimális mérési helyet kapunk. Térjünk rá a szóban forgó függvények explicit kifejezésére.

2. Optimális mérési hely meghatározása $n=1$ esetben

Adott tehát egy olyan feladat, amelyben egy konvex függvény értéke és deriváltja két adott pontban ismert. Tegyük fel — az egyszerűség kedvéért —, hogy ez a két függvényérték a 0 és az 1 pontban ismert. A feladat olyan $t\in[0, 1]$ mérési hely meghatározása, ahol a kétoldali maximális eltérés nagyobbika minimális lesz.

Először ehhez meg kell határozni, a mért érték és a maximális lehetséges eltérések közötti függvénykapcsolatokat. A függvénykapcsolatok meghatározása két lépésben történik; először a bal oldali, majd a jobb oldali függvénykapcsolatot határozzuk meg.

A bal oldali maximális eltérés és a mért érték függvénykapcsolatának meghatározása

A most következő 1. ábra szemléletessé teszi mindazokat a megfontolásokat, amelyek alapján a bal oldali maximális eltérés és a mért érték közötti függvénykapcsolatot meghatároztuk. A függvény értéke a [0, 1] intervallum két pontjában ismert. A 0 és 1 közé eső mérési helyet az ábrán t-vel jelöltük. A t mérési hely magasságában felvett mért értéket S jelöli. A függvény B és C pontbeli meredekségét,

vagyis deriváltját az e $(at+b)$ és az f egyenesek reprezentálják. A maximális lehetséges eltérést k_1, a magasságában felvett mérési helyet pedig τ_1 mutatja.

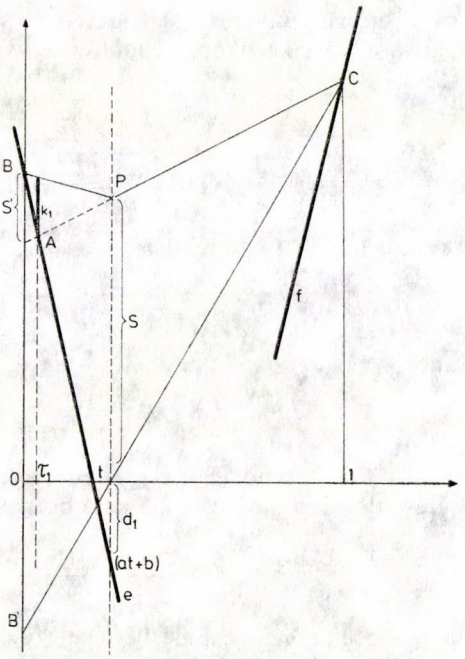
Az ábráról látható, hogy τ_1 (a mért érték) és k_1 (a maximális lehetséges eltérés) hasonlósági transzformáció segítségével egymásba átvihető. Ha az S-sel jelölt szakaszt a C pontból a τ_1-gyel és A-val meghatározott síkra vetítjük, akkor már könnyen látható, hogy S és k_1 egymásba eltolás és nyújtás segítségével transzformálható. Ennek az átvitelnek a segítségével kaphatók meg a következő összefüggések.

Ha S'-t, mint k_1-nek a P pontra vonatkoztatott vetületét vesszük, akkor

$$\frac{k_1}{S'} = \frac{t-\tau_1}{t}. \tag{2.1}$$

A BPD és APF háromszögek hasonlósága miatt

$$\frac{S'}{S+d_1} = \frac{\tau_1}{t-\tau_1}. \tag{2.2}$$



1. ábra

Mivel S és S' között szintén hasonlósági transzformáció írja le a kapcsolatot (S-t az A pontra vonatkozóan az OB egyenesre középpontosan tükrözzük),

$$S' = BB' - c_1 S, \quad \text{ahol} \quad c_1 = \frac{1}{1-t}. \tag{2.3}$$

Ezen három összefüggés figyelembevételével egyszerű számolással adódik a következő függvény

$$k_1 = \frac{(BB' - c_1 S)S}{BB' - S(1-c_1) + d_1} + \frac{d_1}{t}. \tag{2.4}$$

Először meg kell határozni S-nek azt az értékét, amelynél a legnagyobb maximális eltérés (k_1 érték) jelentkezik. Ezt (2.4) S-re vonatkozó szélsőértékének megállapításával kaphatjuk meg. Mielőtt azonban a konkrét számításhoz hozzáfognánk, néhány átalakítást kell elvégeznünk a kapott lineáris törtfüggényen. Az átalakítás azért szükséges, mert noha a kiszámított lineáris törtfüggvény a mért értéket nézve egyértelmű, a mérési helyet tekintetbe véve a függvény jelenlegi alakja tartalmaz olyan paramétereket, amelyek ennek függvényében változnak (d és BB' paraméter).

Fejezzük ki ezeket t függvényében, majd helyettesítsünk vissza. A függvény alakja a következőképpen módosul:

$$k_1 = \frac{\left(OB+\frac{y_0 t}{x_0-t}-\frac{1}{1-t}S\right)S}{OB+\frac{y_0 t}{x_0-t}+S\left(1-\frac{1}{1-t}\right)+|at+b|}+\frac{|at+b|}{t}. \tag{2.5}$$

Az így kapott függvényt helyettesítéssel a következő egyszerűbb alakra hozhatjuk

$$k_1 = \frac{\alpha S-\beta S^2}{\alpha+(1-\beta)S+\gamma t}, \tag{2.6}$$

ahol

$$\alpha = OB+\frac{y_0 t}{x_0-t}, \quad \beta = \frac{1}{1-t}, \quad \gamma = \frac{at+b}{t}.$$

Habár a függvény alakja így lényegesen egyszerűbb, a legnagyobb maximális eltérést előidéző mért értékre kissé bonyolult eredmény adódik:

$$S_{1,2} = \frac{-2\beta(\alpha+\gamma t)\pm\sqrt{4\beta^2(\alpha+\gamma t)^2+4\alpha(1-\beta)(\alpha+\gamma t)}}{2\beta(1-\beta)}. \tag{2.7}$$ [1]

Megkaptuk tehát azt a mért értéket (függvényértéket), amelynél bal oldalon a konvex függvény gráfjára vonatkozó bizonytalanság a legnagyobb.

A jobb oldali maximális eltérés és a mért érték függvénykapcsolatának meghatározása.

Az ábrából kitűnik, hogy a jobb oldali függvénykapcsolat meghatározásának menete nem különbözhet sokban a bal oldaliétól, mindössze a paraméterekben van eltérés. Az ábra alapján a következő összefüggések írhatók fel:

$$\frac{k_2}{S''} = \frac{\tau_2-t}{1-t}, \quad \frac{S''}{S+d_2} = \frac{1-\tau_2}{\tau_2-t}, \tag{2.8}$$

$$S'' = CC'-c_2 S, \quad c_2 = \frac{1}{t}.$$

Ezekből az összefüggésekből a maximális eltérésre, mint függő változóra és a mért értékre, mint független változóra a következő függvény adódik:

$$k_2 = \frac{S(CC'-c_2 d_2)-\frac{1}{t}S^2+CC'd_2}{CC'+S(1-c_2)+d_2}. \tag{2.9}$$

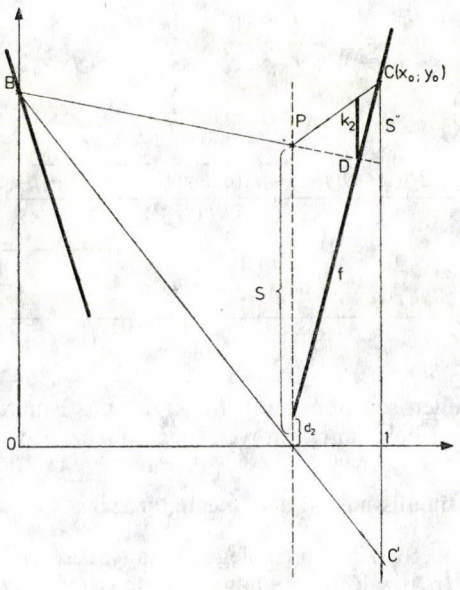
[1] A függvény, amelynek zérushelyeit megkaptuk, olyan alakú racionális törtfüggvény, amely minimumát a + jelhez tartozó gyöknél veszi fel.

Helyettesítés segítségével a következő egyszerűbb alak adódik:

$$k_2 = \frac{\delta S - \frac{1}{t} S^2}{\left(1 - \frac{1}{t}\right) S + \delta + \zeta}, \tag{2.10}$$

ahol

$$\delta = \frac{y_1}{t} + y_1 - \frac{1}{t}|et+f| \quad \zeta = \left(1 + \frac{1}{t}\right)|et+f|.$$



2. ábra

A függvény átalakítása során a mért értéktől nem függő paraméterek, amelyek a mérési helyet tekintve változók, t függvényében lettek kifejezve. Az előbb felírt függvényalak segítségével a legnagyobb maximális eltéréshez tartozó mért érték már könnyen meghatározható:

$$S_{1,2} = \frac{\frac{2}{t}(\delta+\zeta) \pm \sqrt{\frac{4}{t^2}(\delta+\zeta)^2 + 4\delta^2(1-\zeta)\frac{1}{t}\beta}}{-\frac{2}{t}\beta}. \tag{2.11}$$

Megkaptuk tehát mindkét oldalon a legnagyobb maximális eltéréshez tartozó mért értéket. Nyilvánvaló, hogy ezek a mért értékek általában nem egyeznek meg. Bizonyítható viszont, hogy a mérési hely akkor optimális, ha a két oldalon vett legnagyobb maximális eltérés megegyezik. Helyettesítsük tehát vissza minkét oldali függvénynél a legnagyobb maximális eltéréshez tartozó maximális eltérést. A két

függvényt egyenlővé téve, az egyenletet megoldva megkapjuk az optimális mérési helyet. Írjuk fel most az egyenletet:

$$(\alpha-\beta K)K+\gamma=\frac{L\delta-\frac{1}{t}L}{\frac{y_1}{t}+y_1+\eta L+|et+f|}, \tag{2.12}$$

ahol

$$\alpha=OB+\frac{y_0 t}{x_0-t},\quad \beta=\frac{1}{1-t},\quad \gamma=\frac{|at+b|}{t},$$

$$\delta=\frac{y_1}{t}+y_1-\frac{1}{t}|et+f|,\quad \zeta=1+\frac{1}{t}|et+f|,$$

$$\eta=1-\frac{1}{t},$$

$$K=\frac{-2\beta(\alpha+\gamma t)+\sqrt{4\beta^2(\alpha+\gamma t)^2+4\alpha\beta(1-\beta)(\alpha+\gamma t)}}{2\beta(1-\beta)},$$

$$L=\frac{\frac{2}{t}(\delta+\zeta)+\sqrt{\frac{4}{t^2}(\delta+\zeta)^2+4\delta^2(1-\zeta)\frac{1}{t}\eta}}{-\frac{2}{t}\eta}.$$

Bár az egyenlet meglehetősen bonyolult, t-n kívül más ismeretlent nem tartalmaz és így az optimális mérési hely már könnyen meghatározható.

3. Optimális mérési hely meghatározása $n=2$ esetben

A számolás jelen esetben is és tetszőlegesen nagy n esetén is a különböző (τ_i, τ_{i+1}) részintervallumokon realizálódó legnagyobb maximális eltérések egyenlőségén alapul. Ha ennek a feltételnek megfelelően helyezünk el adott számú mérési helyet, akkor az intervallum két végpontjában ismert konvex függvényről a mérési helyek számát tekintve maximális információt szerzünk. (Feltéve, hogy a méréseket *egyszerre* végezzük, azaz passzív algoritmusról van szó.)

3.1. LEMMA. Optimális megoldás tetszőleges n esetén a mérési helyek olyan eloszlása mellett létezik, amely mellett a részintervallumokra vonatkozó maximális lehetséges eltérések egyenlők. (Optimális pontelhelyezés létezése a 3.2. lemmából következik.)

Bizonyítás: Tekintsünk egy tetszőleges $[a, b]$ intervallumot, amelyen a mérési helyek a $\tau_1, \tau_2, \ldots, \tau_n$ pontokban helyezkednek el. Nézzük meg, hogy két szomszédos mérési hely elhelyezkedésétől (τ_i, τ_{i+1}) hogyan függenek a maximális lehetséges eltérések (ξ_i-k).

Tegyük fel az 1.1. lemma alapján, hogy $n=1$ esetben van egy olyan τ optimális mérési hely, hogy $\xi_1(\tau)=\xi_2(\tau)=\inf\max(\xi_1, \xi_2)$. Tekintsük most $n=2$ esetben egy $[a, b]$ intervallumot, amelyen két mérési helyet (τ_1, τ_2) kell optimálisan elhelyeznünk.

Rögzítsük először τ_1-et és ezen feltétel mellett adjuk meg τ_2 optimális értékét. Az 1.1. lemma alapján

$$\max\{\max[\xi_1(a,\tau_1),\xi_2(\tau_1,\tau_2)],\xi_3(\tau_2,b)\}$$

akkor lesz minimális τ_2 szerint, ha

$$\max[\xi_1(a,\tau_1),\xi_2(\tau_1,\tau_2)]=\zeta_3(\tau_2,b). \tag{3.1}$$

Ha τ_2-t rögzítjük, akkor szintén az 1.1. lemma alapján

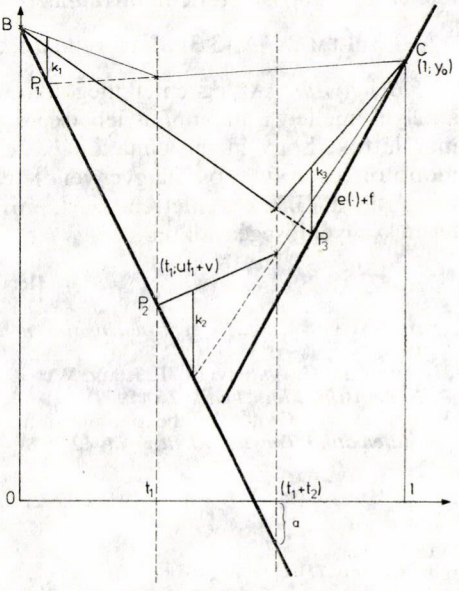
$$\max\{\max[\xi_2(\tau_1,\tau_2),\xi_3(\tau_2,b)],\zeta_1(a,\tau_1)\}$$

akkor lesz minimális, ha

$$\max[\xi_2(\tau_1,\tau_2),\xi_3(\tau_2,b)]=\xi_1(a,\tau_1). \tag{3.2}$$

(3.1)-ből és (3.2)-ből már a lemma állítása közvetlenül következik, tehát beláttuk, hogy a mérési helyek optimális elhelyezéséhez a részintervallumokra vonatkozó maximális lehetséges eltérések egyenlősége szükséges.

Az $n=2$ eset lényegében megegyezik az előbbi esettel, csupán a közbülső $[t_1,(t_1+t_2)]$ intervallum maximális lehetséges eltérésének értelmezése más. Az eltérés nyilván akkor maximális, ha a t_1 és (t_1+t_2) pontokban vett mért értékek egyenlők. Ezen mért értékek együttes csökkentésével a maximális lehetséges eltérés növekszik. Így viszont feltétlenül elérünk az érintők egyikéhez, amely alá a feltételek értelmében a mért érték nem csökkenhet. Így a $[t_1,(t_1+t_2)]$ intervallum két végpontjában felvett mért értékek közül az egyiknek feltétlenül az érintő határegyenesek egyikére kell esni az optimalizálás során.



3. ábra

A korábban tárgyalt két ábra összefüggései alapján $n=2$ esetre a legnagyobb maximális eltérések és a mért értékek között a következő függvénykapcsolatok állíthatók fel

$$k_1=\frac{(BB'-c_1S_1)S_1}{BB'-S_1(1-c_1)+a}+\frac{a}{t_1},$$

ahol

$$BB'=OB+\frac{y_0t_1}{1-t_1},\quad c_1=\frac{1}{1-t_1},$$

$$a=|ut_1+v|;$$

$$k_2=\frac{(DD'-c_2S_2)}{DD'-S_2(1-c_2)+b}+\frac{b}{t_1+t_2}, \tag{3.3}$$

ahol

$$DD' = ut_1+v+y_0+(y_0-t_1-t_2)(t_1-1),$$

$$c_2 = \frac{1}{1-t_1-t_2}, \quad b = u(t_1+t_2)+v;$$

$$k_3 = \frac{S_3(CC'-c_3d)-c_3S^2+CC'd}{CC'+S_3(1-c_3)+b},$$

ahol

$$c_3 = \frac{1}{t_1+t_2}, \quad b = e(t_1+t_2)+f, \quad CC' = y_0+u+v.$$

A fenti függvények szélső értékét S_1-re, S_2-re és S_3-ra meg kell határozni, majd a behelyettesítés után a 3.1. lemma alapján a maximális eltérések egyenlőségét feltételezve a két ismeretlent tartalmazó egyenletrendszert megoldani.

3.2. LEMMA. A (3.3) alatti rendszernek mindig létezik megoldása.

Bizonyítás: Az egyenlet megoldása olyan t_1 és (t_1+t_2) értékek mellett létezik, amelyek mellett a maximális lehetséges eltérések egyenlők. Az 1.1. és 3.1. lemmánál már láttuk, hogy mivel a maximális lehetséges eltérések t_1-nek, illetve (t_1+t_2)-nek monoton és folytonos függvényei, léteznek olyan t_1 és (t_1+t_2) értékek, amelyek kielégítik a (3.3) egyenletrendszert, mivel az egyenletek felállítása a maximális eltérések egyenlőségén alapul.

IRODALOM

[1] BAHVALOV, N. SZ., *A gépi matematika numerikus módszerei* (Műszaki Könyvkiadó, Budapest, 1977).
[2] MICHELLI, C. A., RIVLIN, T. J. and WINOGRAD, S., "The optimal recovery of smoth functions", *Numerische Mathematik* **26** (1979) 191—200.
[3] SONNEVEND, GY., "On the optimization of adaptive algorithms", *ELTE Numerikus és Gépi Matematika Tanszék Kiadványai*, 12/1978.

(Beérkezett: 1979. október 5.)

FEUER GÁBOR
1056 BUDAPEST, BELGRÁD RKP. 17.

A NUMERICAL METHOD FOR THE BEST APPROXIMATION OF A CONVEX FUNCTION

G. FEUER

This paper contains a new numerical method, which seeks to obtain the best approximations of elements of a class of convex functions, supposing that the approximations should be based on the knowledge of the values of the function at n points chosen by us. The method (i.e. its class) presupposes information about the derivatives at the two final points. Having the conditions of the optimization, our aim is to minimize (on an interval) the uncertainity concerning the function values by computing simultaneously n new arguments in the interval, in order that, having at this points the values of the function, we could diminish over a given class as much as possible, the uncertainity concerning the graph of the function on the interval.

Alkalmazott Matematikai Lapok 6 (1980) 83—92

EGY GYORS NORMÁLIS VÉLETLENSZÁM GENERÁTOR

DEÁK ISTVÁN

Budapest

Több algoritmus ismeretes normális eloszlású pszeudo-véletlen számok gyors előállítására. Ezek hátránya a nagyszámú konstans tárolásának és bonyolult gépi kódos program írásának a szükségessége. Az általunk közölt algoritmus egy KINDERMAN és RAMAGE által javasolt algoritmus javított változata. Előnye a viszonylag egyszerű program és a kisszámú konstans.

1. Bevezetés

Normális eloszlású pszeudo-véletlen számok generálására szimulációs számítások folyamán gyakran van szükség. Ilyen véletlenszám generátorokat nagy számban javasoltak már, összehasonlításukról lásd például PAYNE [9] összefoglaló cikkét. Eszerint és más számítógépes futások szerint ([1], [2]) két algoritmust tekinthetünk a leggyorsabbnak: a MARSAGLIA által javasolt RWT algoritmust [8] és az AHRENS és DIETER által közölt FL_5 algoritmust [2]. Mindkettő biteken végzendő műveletek programozását is igényli, a hatékony számítógépes program így csak gépi kódban írható meg. További hátrányuk, hogy sok állandó tárolására van szükség.

Az általunk [4]-ben javasolt algoritmus alapja kettős: részint egy KINDERMAN és RAMAGE [5] által kidolgozott normális generátor, részint a [4]-ben közölt takarékos véletlenszám generálási eljárás. A továbbiakban ezekre részletesen kitérünk, illetőleg megadjuk az új normális generátor algoritmusát a szükséges konstansokkal együtt. Az új algoritmus kevés állandót használ, gyorsasága lényegében azonos az RWT és az FL_5 algoritmusokéval. A *Kinderman—Ramage-algoritmusban* szükséges 2,1 számú egyenletes eloszlású véletlen szám helyett az új algoritmusban $N=1{,}6$ egyenletes eloszlású véletlen szám kell csak egy normális szám generálásához. Eredményeinket a [4] cikkben már lényegében közöltük, itt az algoritmus részletes leírását adjuk.

2. Egy takarékos véletlenszám generálási módszer

Az általánosan ismert elfogadás—elvetés módszer a következőképpen írható le. Legyen feladatunk $f(x)$ sűrűségfüggvényű véletlen számok előállítása és tegyük fel, hogy rendelkezésünkre áll egy másik $h(x)$ sűrűségfüggvény, amelyre az

$$f(x) \leqq ch(x), \quad -\infty \leqq x \leqq +\infty \tag{2.1}$$

egyenlőtlenség fennáll valamilyen $c \geqq 1$ konstans esetén. Ekkor $f(x)$ sűrűségfüggvényű valószínűségi változókat állíthatunk elő a következő algoritmus segítségével.

Elfogadás—elvetés algoritmus

1. Generáljuk x-et h sűrűséggel és u-t egyenletesen.
2. [Ellenőrzés.] Ha $f(x)/ch(x) \leqq u$, akkor elvetjük x-et és megismételjük az eljárást az 1. lépéstől.
3. Adjuk át x-et mint f sűrűségfüggvényűt.

A 2. lépésben néhány generált értéket elvetünk bizonyos valószínűséggel. Azokat az értékeket, melyeket az átlagosnál gyakrabban vetünk el, felesleges értékeknek, azokat az értékeket pedig, melyeket az átlagnál ritkábban vetünk el, hiány értékeknek fogjuk nevezni. Egy olyan transzformációt keresünk a továbbiakban, amely egyes felesleges értékeket oda transzformál, ahol hiány értékek vannak.

A módszer megfogalmazásában nem törekszünk teljes általánosságra. Legyen f és h két folytonos sűrűségfüggvény és legyen

$$I^+ = \{x | h(x) > f(x)\},$$

$$I^- = \{x | h(x) \leqq f(x)\},$$

ahol I^+ a felesleg tartomány, I^- a hiány tartomány. Definiáljuk a

$$g^+(x) = \begin{cases} h(x) - f(x), & \text{ha } x \in I^+, \\ 0, & \text{egyébként,} \end{cases} \tag{2.2}$$

$$g^-(x) = \begin{cases} f(x) - h(x), & \text{ha } x \in I^-, \\ 0, & \text{egyébként,} \end{cases}$$

$$T: I^+ \to I^-$$

függvényeket. Legyen T egy olyan transzformáció, amely kielégíti a következő egyenletet

$$\int_{-\infty}^{x} g^+(y)\,dy = \int_{-\infty}^{T(x)} g^-(y)\,dy. \tag{2.3}$$

Mivel mindkét integrál x-nek, illetőleg $T(x)$-nek monoton növő függvénye, ezért $T(x)$-et a (2.3) egyenlőség meghatározza. A definíció egyértelművé tehető például a

$$T(x) = \min\left\{t \,\middle|\, \int_{-\infty}^{x} g^+(y)\,dy = \int_{-\infty}^{t} g^-(y)\,dy\right\} \tag{2.4}$$

képlet segítségével. A megfelelő véletlenszám generálási algoritmus a következő.

Takarékos módszer

1. Generáljuk x-et h sűrűség szerint.
2. Ha $x \in I^-$, akkor adjuk át x-et.
3. Generáljuk u-t. Ha $f(x)/h(x) \leqq u$, akkor transzformáljuk ezt a felesleg értéket a hiány tartományba T-vel, azaz adjuk át a $T(x)$ értéket, egyébként adjuk át x-et.

Bebizonyítjuk, hogy a fentebbi eljárás valóban f sűrűségfüggvényű véletlen számokat állít elő. Jelöljük y-nal az algoritmus által előállított értéket, legyen u egyenletes [0, 1)-ben, x pedig egy h sűrűségfüggvényű valószínűségi változó. Ekkor

$$(2.5)\qquad P\{y \leqq a\} = P\{x \leqq a,\ x \in I^-\} + P\{x \leqq a,\ x \in I^+,\ f(x)/h(x) > u\} +$$

$$+ P\{T(x) \leqq a,\ x \in I^+,\ f(x)/h(x) \leqq u\} =$$

$$= \int\limits_{\substack{-\infty \\ x \in I^-}}^{a} h(x)\,dx + \int\limits_{\substack{-\infty \\ x \in I^+}}^{a} f(x)\,dx + \int\limits_{\substack{-\infty \\ x \in I^+}}^{T^{-1}(a)} \int\limits_{f(x)/h(x)}^{1} h(x)\,du\,dx.$$

Az utolsó integrál a (2.3) egyenlőség felhasználásával

$$\int\limits_{\substack{-\infty \\ x \in I^+}}^{T^{-1}(a)} (h(x) - f(x))\,dx = \int\limits_{-\infty}^{T^{-1}(a)} g^+(x)\,dx = \int\limits_{-\infty}^{a} g^-(x)\,dx.$$

Ezt az integrált pedig (2.5)-be helyettesítve megkapjuk a keresett eredményt.

Egyszerűség kedvéért tegyük fel, hogy egy h sűrűségfüggvényű valószínűségi változót egy egyenletes eloszlású szám felhasználásával elő lehet állítani. Ekkor a takarékos módszer által előállított f sűrűségfüggvényű valószínűségi változó egy realizációjához szükséges egyenletes eloszlású valószínűségi változók számának várható értéke.

$$N = \int\limits_{\substack{-\infty \\ x \in I^-}}^{+\infty} h(x)\,dx + 2 \int\limits_{\substack{-\infty \\ x \in I^+}}^{+\infty} h(x)\,dx = 1 + \int\limits_{\substack{-\infty \\ x \in I^+}}^{+\infty} h(x)\,dx,$$

ez pedig legfeljebb 2. Az elfogadás—elvetés módszernél $N = 2c \geqq 2$. Módszerünk a WALKER [10] által diszkrét eloszlásokra adott algoritmus általánosításának is tekinthető, de ott $N = 2$. Megjegyezzük még, hogy a T transzformációt az esetek $100 \int\limits_{-\infty}^{+\infty} g^+(x)\,dx$ százalékában kell csak végrehajtani.

Vezessünk be néhány megszorítást a sűrűségfüggvényekre. Az általános eset — a legtöbb gyakorlati szempontból érdekes eloszlásra — ilyen egyszerű esetek összetevéseként tárgyalható. Legyen $f(x)$ és $h(x)$ két sűrűségfüggvény az $[a, b]$ intervallumban, F és H pedig a megfelelő eloszlásfüggvények. Tegyük fel, hogy

$$f(x) < h(x), \quad x \in [a, c] \quad \text{és} \quad f(x) \geqq h(x), \quad x \in [c, b]$$

igaz egy $c \in [a, b]$ konstansra. Jelöljük most $U(x)$-szel a (2.3) egyenlőség által meghatározott T transzformációt, $V(x)$-szel pedig a

$$\int\limits_{x}^{c} g^+(y)\,dy = \int\limits_{c}^{V(x)} g^-(y)\,dy$$

mérlegegyenlőség által adott transzformációt. Ekkor

$$\int_a^x (h(y)-f(y))\,dy = \int_c^{U(x)} (f(y)-h(y))\,dy, \tag{2.6}$$

$$\int_x^c (h(y)-f(y))\,dy = \int_c^{V(x)} (f(y)-h(y))\,dy. \tag{2.7}$$

Ezekből a $K(x)=F(x)-H(x)$ jelölést használva kapjuk, hogy

$$K(U(x)) = K(c)-K(x), \tag{2.8}$$

$$K(V(x)) = K(x). \tag{2.9}$$

A takarékos módszer gyakorlati alkalmazhatósága a K függvény könnyű invertálhatóságán múlik.

A $T(x)$ által adott értékeket WALKER „*alias*"-nak nevezi. Míg diszkrét eloszlásokra könnyű meghatározni a T transzformációt (például az egyenletes eloszláshoz), addig általában, folytonos sűrűségfüggvény esetén ez elég nagy problémát okozhat. A numerikus invertálás lehetősége azonban mindig fennáll. Megemlítjük még, hogy a módszer nehézség nélkül kiterjeszthető véletlen vektorok generálására.

3. Háromszög alakú sűrűségfüggvény

Egyetlen példát adunk itt most a takarékos módszer alkalmazására; háromszög vagy trapéz alakú sűrűségfüggvények szerinti véletlen minta előállításával fogunk foglalkozni. (További példák [4]-ben.)

Legyen $f(x)=2x$, $x\in[0, 1]$ és $h(x)=1$, $x\in[0, 1]$ az egyenletes eloszlás sűrűségfüggvénye. Némi számolás után a (2.8) és a (2.9) egyenlőségből

$$U(x)= 0{,}5+\sqrt{x-x^2}$$

$$V(x) = 1-x.$$

A $V(x)$ transzformációt felhasználó, $f(x)$ sűrűségfüggvényű valószínűségi változót generáló eljárás a következő.

1. Generáljuk u-t.
2. Ha $u>0{,}5$, akkor adjuk át u-t.
3. Generáljuk v-t. Ha $v<2u$, akkor adjuk át u-t, egyébként az $1-u$ értéket.

Az $f(x)$ háromszög alakú sűrűségfüggvényt követő véletlen minták generálására több módszer ismert, (i) az inverziós módszer, amelyben az $x=\sqrt{u}$ számot állítjuk elő, (ii) a rendezett minták elméletén alapuló módszer, amely az $x=\max(u_1, u_2)$ értéket számítja ki, (iii) és az $x=1-|u_1+u_2-1|$ összefüggéssel mintát előállító eljárás.

Ha az általunk fentebb javasolt módszert ezzel a három eljárással összehasonlítjuk, rögtön látszik, hogy eljárásunk kevesebb időt igényel, mint ezek, ugyanis (ii) és (iii) két egyenletes számot használ mintánként, míg eljárásunk átlagosan csak másfelet, valamint a négyzetgyökvonás (i)-ben körülbelül háromszor annyi időt igényel, mint egy egyenletes eloszlású szám generálása.

4. Az ET normális generátor

A KINDERMAN és RAMAGE [5] által javasolt KR algoritmus az összetevéses módszeren alapszik. Legyen adva a standard normális sűrűségfüggvény a következőképpen

$$\varphi(x) = (2\pi)^{-1/2} \exp\{-x^2/2\} = q_1 f_1(x) + \ldots + q_5 f_5(x). \tag{4.1}$$

A q_i valószínűségeket és az f_i sűrűségfüggvényeket az 1. táblázat tartalmazza, f_i a megadott intervallumon kívül zérusnak veendő, a c_i konstansokat pedig az $\int_{-\infty}^{+\infty} f_i(x)\,dx = 1$ egyenlőség meghatározza.

Lényegében véve a (4.1) egyenlőség a $\varphi(x)$ sűrűségfüggvény alatti részt a következő öt részre osztja: egy nagy háromszögre $q_1 f_1(x)$, $x \in [0, b]$, három kis, majdnem háromszög alakú maradékra $q_2 f_2$, $q_3 f_3$, $q_4 f_4$ és végül az eloszlás végére (a konstansok pontos értékét a 3. táblázatban adjuk meg).

1. TÁBLÁZAT

A keverék sűrűségfüggvényei és valószínűségei

i	Valószínűség q_i	Intervallum $a_i \leqq \lvert x\rvert \leqq b_i$	Sűrűségfv. $f_i(x)$
1	0,884	0,00 2,22	$c_1(2{,}22 - \lvert x\rvert)$
2	0,027	0,00 0,48	$c_2(\varphi(x) - q_1 f_1(x))$
3	0,047	0,48 1,59	$c_3(\varphi(x) - q_1 f_1(x))$
4	0,015	1,59 2,22	$c_4(\varphi(x) - q_1 f_1(x))$
5	0,027	2 22 $+\infty$	$c_5 \varphi(x)$

Az eredeti KR algoritmuson három módosítást hajtottunk végre. Mindenekelőtt az f_1 sűrűségfüggvény szerinti mintavételre a takarékos módszert használtuk. Másodszor, az u döntési változót, amely segítségével az f_i sűrűségfüggvényt választottuk, minden lépésben transzformáltuk úgy, hogy a [0, 1) intervallumban újra egyenletes legyen; ezzel tovább csökkentettük a szükséges egyenletes eloszlású véletlen számok számát. Végül az algoritmus lépéseit olyan sorrendbe állítottuk, hogy mindig a nagyobb valószínűségű sűrűségfüggvények felől döntsünk először.

A majdnem háromszög alakú f_2, f_3 és f_4 sűrűségfüggvények szerinti generálást úgy hajtottuk végre, ahogy a KR algoritmusban volt leírva, vagyis a sűrűségfüggvény görbéjét két egymással párhuzamos egyenes közé zártuk (MARSAGLIA [9]). Hasonlóképpen változatlanul meghagytuk az f_5 végből való mintavételi eljárást; ez MARSAGLIA [6] és AHRENS és DIETER [1] ötletein alapszik. Ezekben az esetekben

véleményünk szerint a programozás kényelme fontosabb szempont, mint az elérhető javulás a sebességben vagy a szükséges egyenletes számok várható értékében. A teljesség kedvéért azonban ezeket a módszereket is ismertetjük röviden.

Tegyük fel, hogy egy olyan $g(x)$, $x\in[0, c]$ sűrűségfüggvény szerint akarunk véletlen számokat generálni, amely majdnem háromszög alakú tartományt határol, vagy más szóval majdnem lineáris. Zárjuk a sűrűségfüggvény görbéjét két egymással párhuzamos egyenes közé, legyenek ezek a $b=g(0)$ jelöléssel az $y=-bx/c+a$ és az $y=-bx/c+b$ egyenesek, ahol a egy konstans, $a<b$, valamint $-bx/c+a\leqq \leqq g(x)\leqq -bx/c+b$. Az algoritmus alapötlete a következő: generáljunk egy (x, y) egyenletes eloszlású pontot a majoráló $y=-bx/c+b$ egyenes és a koordinátatengelyek által határolt háromszögben. Ha ez a pont az $y=-bx/c+a$ egyenes alá esik, akkor $g(x)$ alatt van, tehát x-et elfogadjuk (gyors elfogadás), egyébként ismétlünk. Ha az (x, y) pont a két egyenes között van, akkor még mindig megvizsgálandó a $g(x)>y$ egyenlőtlenség, melynek teljesülése esetén szintén elfogadjuk az x számot.

Legyen most u és v két, [0, 1)-ben egyenletes eloszlású szám és legyen $M=\max(u, v)$, $m=\min(u, v)$. Azt állítjuk, hogy az $(m, M-m)$ pont egyenletes eloszlású a (0, 0), (0, 1) és (1, 0) pontok, mint csúcspontok által adott háromszögben. Miután $m+M-m\leqq 1$, ezért az $(m, M-m)$ pont a háromszögben van, a pont $F(x, y)$ eloszlásfüggvényére az $x+y\leqq 1$ feltétel mellett

$$\frac{1}{2}F(x, y)=P\{m<x, M-m<y\}=\frac{1}{2}[P\{m<x, M-m<y|v<u\}\cdot$$

$$\cdot P\{v<u\}+P\{m<x, M-m<y|u<v\}P\{u<v\}].$$

Mivel $P\{v<x, u-v<y|v<u\}=\int_0^x\int_v^{y+v} du\,dv=xy$, ezért $F(x, y)=2xy$, vagyis az $(m, M-m)$ pont koordinátái egyenletes eloszlásúak és függetlenek egymástól.

Az $y=-bx/c+a$ egyenes alá akkor esik a $(cm, b(M-m))$ pont, ha $b(M-m)\leqq \leqq -bm+a$, vagyis $bM\leqq a$. Ezek után a $g(x)$ sűrűségfüggvényű változók előállítására szolgáló algoritmus a következő.

1. Generáljuk u-t és v-t egyenletesen, legyen $M\leftarrow\max(u, v)$, $m\leftarrow\min(u, v)$ $x\leftarrow cm$, $y\leftarrow bM$.
2. Ha $y\leqq a$, akkor adjuk át x-et.
3. Ha $b(M-m)\leqq g(x)$, akkor adjuk át x-et, egyébként ismételjük meg az eljárást az 1. lépéstől.

Legyen most feladatunk olyan x normális eloszlású valószínűségi változók generálása, amelyek valamilyen a értéknél nagyobbak. Más szóval az $f(x)=ce^{-x^2/2}$, $x\in[a, +\infty)$ sűrűségfüggvényű változók generálására keressünk megfelelő módszert, ahol $1/c=\int_a^{+\infty} e^{-x^2/2}\,dx$. Az $f(x)$ függvénynek majoránsa az $\alpha h(x)=\frac{c}{a}xe^{-x^2/2}=$ $=g(x)$ függvény. A módszerünk $f(x)$ sűrűségfüggvényű változók generálására egy elfogadás—elvetés algoritmus, amelyben a fenti $g(x)$ majorálót használjuk. Hasz-

náljuk a

$$G(x) = \int_a^x g(t)\,dt = \frac{c}{a}[e^{-a^2/2} - e^{-x^2/2}]$$

jelölést. Ekkor a $h(x)$ sűrűségfüggvényre

$$h(x) = \frac{g(x)}{G(\infty)} = xe^{-(x^2-a^2)/2}$$

kifejezést kapjuk, a megfelelő eloszlásfüggvény pedig

$$H(y) = \int_a^y h(t)\,dt = \frac{G(x)}{G(\infty)} = 1 - e^{-(x^2-a^2)/2}.$$

Ha most az $1-u=H(y)$ egyenlőségből y-t kifejezzük, akkor h sűrűségfüggvényű változók generálására alkalmas kifejezést nyerünk, ez pedig

$$y = (a^2 - 2\ln u)^{1/2}.$$

Az $f(x)/g(x) < u$ egyenlőtlenség bal oldala a/x lesz a jelen f és g függvényekre, tehát a teljes generálási eljárás $f(x)$ sűrűségfüggvényű változók előállítására a következő.

1. Generáljuk u-t és legyen $x \leftarrow (a^2 - 2\ln u)^{1/2}$.
2. Generáljuk v-t. Ha $a < vx$, akkor menjünk vissza az 1. lépésre.
3. Adjuk át x-et.

Az elfogadás valószínűségét az $1/G(\infty)$ kifejezés adja meg, vagyis ez

$$ae^{a^2/2} \int_a^{+\infty} e^{-x^2}\,dx.$$

A fenti részeket összefoglalva a normális eloszlású változók generálására szolgáló algoritmus végső formája a következő, ahol az

$$f(x) = \varphi(x) - q_1 f_1(x) = d_1 \exp\{-x^2/2\} - d_2(b - |x|)$$

jelölést használtuk, a konstansok értékei pedig a 3. táblázatban találhatók meg.

ET Takarékos háromszög

1. [Nagy háromszög.] Generáljuk u-t. Ha $u > p_1$, akkor menjünk a 2. lépésre, egyébként adjuk át az $x \leftarrow c_{11}u - c_{12}$ értéket.

2. Ha $u > p_2$, akkor menjünk a 4. lépésre, egyébként legyen $x \leftarrow c_{21}u$.

3. Generáljuk v-t, legyen $s \leftarrow v - 0{,}5$, $v \leftarrow |s|$, $t \leftarrow 1 - x$. Ha $v > t$, akkor cseréljük ki x-et t-vel, azaz $x \leftarrow t$. Ha $s < 0$, akkor adjuk át az $x \leftarrow -bx$, egyébként az $x \leftarrow bx$ számot.

4. Ha $u > p_3$, akkor menjünk a 6. lépésre, egyébként legyen $u \leftarrow (u - p_2)c_{41}$.

5. [Második kis háromszög.] Generáljuk v-t. Legyen $z \leftarrow u - v$, $x \leftarrow c_{51} + c_{52} \cdot \min(u, v)$. Ha $z < 0$, akkor változtassuk meg az előjelet, azaz $x \leftarrow -x$. Ha $\max(u, v) \leqq$

$\leqq c_{53}$, akkor adjuk át x-et. Ha $c_{54}|z| \leqq f(x)$, akkor adjuk át x-et, egyébként generáljuk u-t és ismételjük meg ezt a lépést az elejétől.

6. Ha $u > p_4$, akkor menjünk a 8. lépésre, egyébként legyen $u \leftarrow (u - p_3)c_{61}$.

7. [Első kis háromszög.] Generáljuk v-t. Legyen $z \leftarrow u - v$, $x \leftarrow c_{71} - c_{72} \min(u, v)$. Ha $z < 0$, akkor változtassuk meg az előjelet, azaz $x \leftarrow -x$. Ha $\max(u, v) \leqq c_{73}$ akkor adjuk át x-et. Ha $c \leftarrow |z| \leqq f(x)$, akkor adjuk át x-et, egyébként generáljuk u-t és ismételjük meg ezt a lépést elölről.

8. Ha $u > p_5$, akkor menjünk a 10. lépésre, egyébként legyen $u \leftarrow (u - p_4)c_{81}$.

9. [Harmadik kis háromszög.] Generáljuk v-t, legyen $z \leftarrow u - v$, $x \leftarrow b - c_{91} \cdot \min(u, v)$. Ha $z > 0$, akkor változtassuk meg az előjelet, azaz $x \leftarrow -x$. Ha $\max(u, v) \leqq c_{92}$, akkor adjuk át x-et. Ha $c_{93}|z| \leqq f(x)$, akkor adjuk át x-et, egyébként generáljuk u-t és ismételjük meg ezt a lépést elölről.

10. [Vég.] Számítsuk ki az $u \leftarrow (u - p_5)c_{101}$ számot.

11. Generáljuk v-t, legyen $s \leftarrow v - 0{,}5$, $v \leftarrow 2|s|$, $t \leftarrow c_{111} - \ln v$. Ha $u^2 t > c_{111}$, akkor generáljuk u-t és ismételjük meg ezt a lépést az elejétől, egyébként adjuk át az $x \leftarrow (2t)^{1/2}$ számot, ha $s > 0$, illetőleg az $x \leftarrow -(2t)^{1/2}$ számot, ha $s > 0$.

A szükséges egyenletes eloszlású véletlen számok N várható értékét a következő módon lehet meghatározni. Az f_1 sűrűség szerinti generáláshoz 1,5 egyenletes szám kell. A három kis háromszög alakú sűrűségfüggvény esetén az együttes elfogadási valószínűség 0,827, a végben pedig 0,849. Ezek szerint:

$$N = 0{,}844 \cdot 1{,}5 + 0{,}0892(2/0{,}827) + 0{,}0267(2/0{,}849) = 1{,}604.$$

5. Számítógépes tapasztalatok és összehasonlítások

A számítógépes algoritmusokat a KSH ÁSzSz *Honeywell Bull* 66/60 számítógépén futtattuk, melynek ciklusideje körülbelül 1 μsec. A kapott futási idők mintegy 2—4%-os szóródást mutattak. A futások időadatai és a memóriaigények a 2. táblázatban találhatók.

2. TÁBLÁZAT

Egyenletes és normális számok generálásának ideje (μsec)
F FORTRANT-t, G gépi kódot jelent

	N	Egyenletes, dupla pontosság F	Egyenletes, szimpla pontosság F	Egyenletes, eltolások G	*Memória*	
					sor	szó
Egyenletes	—	74	52	25	—	—
PO F	1,27	230	208	174	17	55
KR F	2,16	214	169	105	74	271
ET F	1,60	180	149	104	79	308
ET G	1,60	—	—	63	—	278
FL_5 G	1,23	—	—	60	—	255

Abból a célból, hogy az algoritmusok különböző viselkedését demonstrálni tudjuk, szándékosan három különböző egyenletes generátort használtunk: egy lassút, egy közepest és egy nagy sebességűt. Az első egy kongruenciális generátor volt:

$u_i = k_i/m$, ahol $k_{i+1} = k_i a \pmod m$ a $k_0 = 2{,}910\,383 \cdot 10^{-11}$, $a = 53\,088\,871\,541$, $m = 2^{35}$ értékeket AHRENS és DIETER [3] könyvéből (1—6. old.) vettük, melyben a szorzást dupla pontossággal hajtottuk végre. A második generátor csak abban különbözött az elsőtől, hogy a szorzást szimpla pontossággal hajtottuk végre. A harmadik generátor egy gépi kódban (GMAP) megírt szubrutin volt, amely egy egyenletes eloszlású számot két eltolással és egy összeadással állított elő. N-nel jelöltük a szükséges egyenletes eloszlású számok számának várható értékét. A táblázat utolsó oszlopában a memóriaigényeket adtuk meg FORTRAN sorokban és szavakban; a szükséges konstansok helyigényét beleértve, de az external rutinokét (exp, sqrt) kizárva.

Az „Egyenletes" sor az éppen feltüntetett generátor idejét tartalmazza. PO a polár módszert jelenti, ahogyan [1]-ben van leírva, KR az eredeti *Kinderman—Ramage algoritmus* [5], aztán szerepel az ET algoritmus FORTRAN, illetőleg gépi kódos algoritmusa, végül AHRENS és DIETER FL_5 algoritmusa van megadva [2].

Amint az várható volt, az ET algoritmus KR-nél gyorsabbnak bizonyult, bár gyorsabb véletlenszám generátorokra a különbség egyre kisebb a FORTRAN-ban írt változatok sebessége között (az egyenletes eloszlású minták számának csökkenését a némileg hosszabb program kiegyensúlyozza). Az ET algoritmust FL_5-höz hasonlítva látható, hogy lényegében azonos a futási idejük és memóriaigényük. A következő eredményt idézzük most. AHRENS és DIETER [3] könyvében (13—14. old.) található a normális generátorok egy összehasonlítása. A két leggyorsabb az FL_5 és MARSAGLIA RWT eljárása volt, mindkettőnek csak kevéssel volt kevesebb a futási ideje egy négyzetgyökvonáshoz szükséges időnél. Miután a HWB 66/60 gépen a négyzetgyökvonás ideje 88 μsec, így mind ET, mind FL_5 futási ideje ennél jóval kisebb. Így mondhatjuk, hogy az ET algoritmus ekvivalens a leggyorsabb ismert normális generátorokkal.

Az ET javára szól, hogy csak kisszámú konstans (27) tárolását igényli, valamint az, hogy nincs szükség eltolásra vagy másmilyen bit manipulációra (az esetleges előjel meghatározást kivéve). Ellene szól a viszonylag hosszú program, valamint az exponens és a négyzetgyökvonást végző külső rutinok használata.

Végül két megjegyzést teszünk. Véleményünk szerint másmilyen, háromszög vagy trapéz alakú sűrűségfüggvény szerint generálást végző algoritmusok a fentiekhez hasonlóan megjavíthatók a takarékos módszer alkalmazásával (lásd például a TR algoritmust [1]). Másodszor, a konstansok egy részét [5]-ből vettük, más részét a következő kifejezések segítségével számítottuk ki:

$$p_1 = p_2/2, \quad p_2 = b/\sqrt{2\pi}, \quad c_{11} = b/p_1, \quad c_{12} = b/2,$$
$$c_{21} = 1/p_2, \quad c_{41} = 1/(p_3 - p_2), \quad c_{61} = 1/(p_4 - p_3),$$
$$c_{81} = 1/(p_5 - p_4), \quad c_{101} = 1/(1 - p_5), \quad c_{111} = b^2/2,$$
$$d_1 = 1/\sqrt{2\pi}, \quad d_2 = d_1/b.$$

3. TÁBLÁZAT

Az ET algoritmusban használt konstansok

b =	2,21 603	58 671	66 471
p_1 =	0,44 203	52 011	49 379
p_2 =	0,88 407	04 022	98 758

$$
\begin{aligned}
p_3 &= 0{,}93\,147\ 84\,468\ 00\,518\\
p_4 &= 0{,}95\,872\ 08\,247\ 90\,463\\
p_5 &= 0{,}97\,331\ 09\,541\ 73\,898\\
c_{11} &= 5{,}01\,325\ 65\,492\ 62\,003\\
c_{12} &= 1{,}10\,801\ 79\,335\ 83\,235\\
c_{21} &= 1{,}13\,113\ 16\,354\ 44\,180\\
c_{41} &= 21{,}09\,346\ 65\,310\ 41\,488\\
c_{51} &= 0{,}47\,972\ 74\,042\ 22\,441\\
c_{52} &= 1{,}10\,547\ 36\,610\ 22\,070\\
c_{53} &= 0{,}87\,283\ 49\,766\ 71\,790\\
c_{54} &= 0{,}04\,926\ 44\,963\ 73\,128\\
c_{61} &= 36{,}70\,751\ 50\,476\ 61\,921\\
c_{71} &= 0{,}47\,972\ 74\,042\ 22\,441\\
c_{72} &= 0{,}59\,550\ 71\,380\ 15\,940\\
c_{73} &= 0{,}80\,557\ 79\,244\ 23\,817\\
c_{74} &= 0{,}05\,337\ 75\,495\ 06\,886\\
c_{81} &= 68{,}53\,948\ 81\,511\ 71\,341\\
c_{91} &= 0{,}63\,083\ 48\,019\ 21\,960\\
c_{92} &= 0{,}75\,559\ 15\,316\ 67\,601\\
c_{93} &= 0{,}03\,424\ 05\,037\ 50\,111\\
c_{111} &= 37{,}46\,855\ 56\,956\ 85\,297\\
c_{111} &= 2{,}45\,540\ 74\,822\ 84\,127\\
d_1 &= 0{,}39\,894\ 22\,804\ 01\,433\\
d_2 &= 0{,}18\,002\ 51\,910\ 68\,563
\end{aligned}
$$

IRODALOM

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

[2] AHRENS, J. H. and DIETER, U., "Extensions of *Forsythe's method* for random sampling from the normal distribution", *Math. Comp.* **27** (1973) 927—937.

[3] AHRENS, J. H. and DIETER, U., *Non-uniform Random Numbers,* (Graz, 1974).

[4] DEÁK, I., "An economical method for random number generation and a normal generator", *Computing* (1981) **27** 113—121.

[5] KINDERMAN, A. J. and RAMAGE, J. G., "Computer generation of normal random variables", *J. Amer. Stat. Ass.* **71** (1976) 893—896.

[6] MARSAGLIA, G., "Generating a variable from the tail of the normal distribution", *Technometrics* **6** (1964) 101—102.

[7] MARSAGLIA, G., "Random variables and computers", *Trans. Third Prague Conf. on Inf. Theory, Stat. Dec. Functions,, Prague:* Publishing House of the Czechoslovak Academy of Sciences, 1964, 499—512.

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

[9] PAYNE, W. H., "Normal random numbers: using machine analysis to choose the best algorithm", *ACM TOMS* **3** (1977) 346—358.

[10] WALKER, A. J., "An efficient method for generating discrete random variables with general distributions", *ACM TOMS* **3** (1977) 253—256.

(Beérkezett: 1980. április 22.)

DEÁK ISTVÁN
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST, ÚRI U. 49.

A FAST NORMAL RANDOM NUMBER GENERATOR

I. DEÁK

Several algorithms are known to produce normally distributed pseudo-random numbers at a fast rate. The common disadvantages of these algorithms are the sophisticated assembler coding and the great number of constants. The method presented here is a modified version of an algorithm proposed by KINDERMAN and RAMAGE. The advantage of the method is the relatively simple program and the small number of constants.

Alkalmazott Matematikai Lapok 6 (1980) —104

EGY DISZKRÉT DUPLÁN SZTOCHASZTIKUS FOLYAMATTAL KAPCSOLATOS DÖNTÉSI PROBLÉMÁRÓL

KRÁMLI ANDRÁS, LUKÁCS PÁL, VASSEL RÓBERT

Budapest

Legyen az $\{\eta_n\}$ folyamat 0, 1 értékeket felvevő *Markov-lánc*. A $\{\xi_n\}$ megfigyelhető folyamatot az $\{\eta_n\}$ a következő módon vezérli:

$$P(\xi_n = 1|\eta_n = 0,\ \eta_{n-1}, \ldots, \eta_0, \xi_{n-1}, \ldots, \xi_0) = p_0, \quad P(\xi_n = 1|\eta_n = 1,\ \eta_{n-1}, \ldots, \eta_0, \xi_{n-1}, \ldots, \xi_0) = p_1.$$

Bizonyítjuk, hogy a $\{\xi_n\}$ folyamat múltját fölhasználó bármely döntési eljárás kifejezhető a $\pi_n = P(\eta_n = 0|\xi_n, \ldots, \xi_0)$ a posteriori valószínűségekkel. A $\{\pi_n\}$ folytonos állapotterű *Markov-lánc*, mely a klasszikus értelemben nem ergodikus. A 3.1. tétel a folyamat egy ergodikus tulajdonságát fejezi ki. Egyenletet írunk fel a $\{\pi_n\}$ stacionárius eloszlására, melyet numerikusan megoldunk, és összehasonlítjuk a szimulációval nyert eredményekkel.

1. Bevezetés

A matematikai statisztika egyik alapfeladata a hipotézisvizsgálat, azaz annak eldöntése, hogy adott eloszláscsaládok közül melyikbe tartozik egy valószínűségi változó, melyre több megfigyelést végezhetünk.

A rögzített megfigyelésszám esetén a hipotézis vizsgálat a *Neyman—Pearson-lemma* különböző általánosításai alapján történik, mely szerint egy hipotézis elfogadásának vagy elvetésének átlagos költsége akkor minimális, ha a hipotézist akkor fogadjuk el, amikor a minta a különböző hipotézisekhez tartozó sűrűségfüggvények hányadosainak valamely nívó halmazába esik. Változó megfigyelésszám esetén — szekvenciális módszer — WALD eredményeinek alkalmazásával hasonló módszerek adódnak [2]. A szekvenciális hipotézisvizsgálattal rokon az ún. riasztási feladat, amikor a minta egy véletlen időpontig egy adott eloszlásból származik, majd utána egy másik eloszlásból és feladatunk ennek az időpontnak a jelzése. A költség ebben az esetben a várható késés és a „vaklárma" valószínűségének valamilyen lineáris kombinációja [4].

Általános feltételek mellett igazolták, hogy ebben a feladatban (a szekvenciális döntési eljáráshoz hasonlóan) előnyös a *Bayes-módszer* alkalmazása, aminek lényege abban áll, hogy minden időpontban *Bayes tétele* alapján kiszámítjuk az adott hipotézis igaz voltának a megfigyeléssorozatra vett feltételes valószínűségét — azaz az a posteriori valószínűséget — mert ez a változó összesűríti magában a minta által a hipotézisekre vonatkozó teljes információt.

Több szerző vizsgálta azt az esetet, amikor a hipotézisek maguk is véletlen változók, a legegyszerűbb esetben két hipotézis váltakozik egymással egy időben folytonos, vagy időben diszkrét *Markov-folyamat* szerint. A vizsgált esetben a fenti állítást, ti. hogy a döntéshez elegendő az a posteriori valószínűséget figyelembe venni,

bebizonyították, és rekurzív formulák segítségével leírták az a posteriori valószínűség sztochasztikus viselkedését.

Az a posteriori valószínűség eloszlásfüggvénye folytonos esetben egy integro-differenciálegyenletnek tesz eleget [3].

Az egyenlet analitikus megoldása nem ismeretes — így a numerikus eredmények, táblázatok megadása, különböző módszerek hatékonyságának összehasonlítása szimuláció útján történik.

Dolgozatunkban egy diszkrét modellt tárgyalunk, mely hasonló az [1] dolgozatban tárgyalthoz, numerikus módszerrel meghatározzuk az a posteriori valószínűség határeloszlásának sűrűségfüggvényét, konkrét módszert adunk a döntés költségének minimalizálására, és egy tételt bizonyítunk, miszerint az a posteriori valószínűségek függése a kezdeti paraméterektől exponenciálisan csökken.

2. A modell

Vizsgáljuk a következő feladatot.

Legyen $\eta_0, \eta_1, \eta_2, \ldots, \eta_n$ diszkrét idejű 2 állapotú *Markov-lánc,* melynek átmenet valószínűség mátrixa

$$P(\eta_n = 1|\eta_{n-1} = 0) = P(\eta_n = 0|\eta_{n-1} = 1) = P$$

$$P(\eta_n = 1|\eta_{n-1} = 1) = P(\eta_n = 0|\eta_{n-1} = 0) = 1 - P$$

alakú. A $\xi_0, \xi_1, \xi_2, \ldots, \xi_n$ folymatot, amely szintén a 0 és 1 értékeket veheti fel az $\{\eta_n\}$ folyamat vezérli a következő értelemben:

$$P(\xi_n = 1|\eta_n = 0,\ \eta_{n-1}, \eta_{n-2}, \ldots, \eta_0,\ \xi_{n-1}, \xi_{n-2}, \ldots, \xi_0) = p_0,$$

$$P(\xi_n = 1|\eta_n = 1,\ \eta_{n-1}, \eta_{n-2}, \ldots, \eta_0,\ \xi_{n-1}, \xi_{n-2}, \ldots, \xi_0) = p_1,$$

$$0 < p_0 < 1, \quad 0 < p_1 < 1,$$

tehát a $\{\xi_n = 1\}$ esemény valószínűsége rögzített η_n mellett független az $\{\eta_k\}$ éz $\{\xi_k\}$ folyamatok múltjáról. Megjegyezzük, hogy az $\{\eta_k, \xi_k\}$ folyamatpár *Markov-láac,* de $\{\xi_k\}$ önmagában nem az.

Legyen az $\{\eta_0 = 0\}$ a priori valószínűsége π_0, ekkor jelölje π_n a $P(\eta_n = 0|\xi_n, \ldots, \xi_0)$ a posteriori valószínűséget. Könnyen igazolható a következő lemma

2.1. LEMMA.

$$P(\xi_n, \ldots, \xi_{n+k},\ \eta_n, \ldots, \eta_{n+k}|\pi_{n-1},\ \xi_{n-1}, \ldots, \xi_0) = \tag{2.1}$$

$$= P(\xi_n, \ldots, \xi_{n+k},\ \eta_n, \ldots, \eta_{n+k}|\pi_{n-1}).$$

Bizonyítás.

$$P(\xi_n, \ldots, \xi_{n+k},\ \eta_n, \ldots, \eta_{n+k}|\pi_{n-1},\ \xi_{n-1}, \ldots, \xi_0) =$$

$$= P(\xi_n, \ldots, \xi_{n+k},\ \eta_n, \ldots, \eta_{n+k},\ \{\eta_{n-1} = 0\}|\pi_{n-1},\ \xi_{n-1}, \ldots, \xi_0)\pi_{n-1} + \tag{2.2}$$

$$+ P(\xi_n, \ldots, \xi_{n+k},\ \eta_n, \ldots, \eta_{n+k},\ \{\eta_{n-1} = 1\}|\pi_{n-1},\ \xi_{n-1}, \ldots, \xi_0)(1 - \pi_{n-1}).$$

A $\{\xi_k\}$ és $\{\eta_k\}$ folyamatok definíciója miatt a (2.2) egyenlőség jobb oldali tagjaiban szereplő események függetlenek a feltételtől.

Megjegyzés. A (2.1) összefüggés azt jelenti, hogy π_n a $\{\xi_k\}$ múltjából képzett elégséges statisztika a $\{\xi_k\}$ és $\{\eta_k\}$ jövőjére.

KÖVETKEZMÉNY. A $\{\pi_n\}$ folyamat diszkrét idejű, folytonos állapotterű *Markov-lánc*. A $P(\pi_n|\pi_{n-1})$ feltételes valószínűség *Bayes tétele* alapján határozható meg.

Ha $\xi_n=0$, akkor

$$\pi_n = f_0(\pi_{n-1}) = \left(1+\frac{(1-p_1)(P\pi_{n-1}+(1-P)(1-\pi_{n-1}))}{(1-p_0)((1-P)\pi_{n-1}+P(1-\pi_{n-1}))}\right)^{-1}. \tag{2.3}$$

Ha $\xi_n=1$, akkor

$$\pi_n = f_1(\pi_{n-1}) = \left(1+\frac{p_1(P\cdot\pi_{n-1}+(1-P)(1-\pi_{n-1}))}{p_0((1-P)\pi_{n-1}+P(1-\pi_{n-1}))}\right)^{-1}. \tag{2.3'}$$

Rögzített π_{n-1} érték mellett

$$P(\pi_n = f_0(\pi_{n-1})) =: \varrho(\pi_{n-1}) =$$

$$= (1-p_0)((1-P)\pi_{n-1}+P(1-\pi_{n-1}))+(1-p_1)((1-P)(1-\pi_{n-1})+P\pi_{n-1}),$$

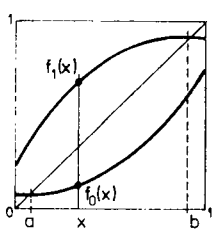
$$P(\pi_n = f_1(\pi_{n-1})) = 1-\varrho(\pi_{n-1}).$$

A 2.1. lemma következtében bármilyen, a $\{\xi_k\}$ folyamat jövőjére vonatkozó, a $\{\xi_k\}$ múltját felhasználó becslési feladat megoldásához elegendő a $\{\pi_k\}$ értékeit figyelembe venni. Így válik szükségessé a $\{\pi_k\}$ *Markov-lánccal* kapcsolatos határeloszlás problémák vizsgálata.

3. Stabilitás

A $\{\pi_k\}$ folyamat a szó klasszikus értelmében nem ergodikus, mert rögzített π_0 értékre a $\pi_1, \ldots, \pi_k, \ldots$ értékkészlete egy megszámlálható halmaz, mely benne van a π_0 és a paraméterek racionális kifejezéseiből alkotott halmazban, ezért pl. egy adott π_k értékhez kontinuum számosságú olyan π érték van, amely biztosan nem lehet a kezdeti π_0. Az irányítási és becslési feladatokban gyakorlati szempontból a szakaszonkénti folytonos és stacionárius (k-tól független) megoldások érdekesek. Ezért felmerül a kérdés, hogyha a $\{\pi_k\}$ folyamat rendelkezik folytonos stacionárius sűrűségfüggvénnyel, akkor az milyen egyenletnek tesz eleget.

Vizsgáljuk meg, hogy egy tetszőleges folytonos kezdeti $h^{(0)}(x)$ sűrűségfüggvény egy lépés után milyen $h^{(1)}(x)$ sűrűségfüggvénybe megy át.



1. ábra

Az 1. ábrából leolvasható, hogy

$$h^{(1)}(y)\,\Delta y \sim h^{(0)}(f_0^{-1}(y))\,\varrho(f_0^{-1}(y))\,\Delta f_0^{-1}(y)+h^{(0)}(f_1^{-1}(y))(1-\varrho(f_1^{-1}(y)))\,\Delta f_1^{-1}(y).$$

Az $f_0^{-1}(y)=:g_0(y), f_1^{-1}(y)=:g_1(y)$ jelölést bevezetve, $\Delta y\to 0$ határátmenetet képezve, és figyelembe véve, hogy $h^{(0)}(x)$ a [0, 1] intervallumon kívül zérus, nyerjük a

$$h^{(1)}(y) = h^{(0)}(g_0(y))\,\varrho(g_0(y))\,g_0'(y)\cdot\chi_{[0,1]}(g_0(y))+ \tag{3.1}$$

$$+h^{(0)}(g_1(y))\cdot(1-\varrho(g_1(y)))\cdot g_1'(y)\cdot\chi_{[0,1]}(g_1(y))$$

összefüggést.

Könnyen látható, hogy a fenti szemléletes levezetés egzakttá tehető a következő jól ismert tény felhasználásával: ha egy ξ valószínűségi változó egy $h(y)$ sűrűségfüggvénnyel rendelkező valószínűségi változó monoton függvénye ($\xi=f(\eta)$), akkor a ξ változónak is létezik a $g(x)$ sűrűségfüggvénye és $g(x)=h(f^{-1}(x))\dfrac{df^{-1}(x)}{dx}$.

A *Stieltjes-integrálra* érvényes $\int_a^b f(x)\,dF(x)=[f(x)\,F(x)]_a^b-\int_a^b F(x)f'(x)\,dx$ összefüggés segítségével levezethető a $H^{(1)}(y)$ és $H^{(0)}(y)$ eloszlásfüggvények közötti

$$H^{(1)}(y) = H^{(0)}(g_0(y))\,\varrho(g_0(y))-\int_0^{g_0(y)}\varrho'(x)H^{(0)}(x)\,dx+$$

$$+H^{(0)}(g_1(y))(1-\varrho(g_1(y)))+\int_0^{g_1(y)}\varrho'(x)H^{(0)}(x)\,dx$$

összefüggés.

A $h^{(0)}(x)=h^{(1)}(x)$, ill. $H^{(0)}(x)=H^{(1)}(x)$ helyettesítéssel a stacionárius sűrűség-, ill. eloszlásfüggvényre nyerünk egyenleteket:

$$h(y) = h(g_0(y))\cdot\varrho(g_0(y))\cdot g_0'(y)\cdot\chi_{[0,1]}(g_0(y))+ \tag{3.1'}$$

$$+h(g_1(y))\cdot(1-\varrho(g_1(y)))\cdot g_1'(y)\cdot\chi_{[0,1]}(g_1(y)).$$

Vegyük észre, hogy az $[a, b]$ intervallum ($a=f_0(a)$, $b=f_1(b)$) „elnyelő" állapot, azaz bármely ε-ra van olyan N, hogy tetszőleges π_0-ra $a-\varepsilon<\pi_n<b+\varepsilon$, ha $n\geqq N$. Ezért a $h(y)$ stacionárius sűrűségfüggvény — amennyiben létezik — az $[a, b]$ intervallumra koncentrálódik. Igazolható viszont a következő állítás; ha valamely rögzített π_0 mellett létezik a $\lim_{n\to\infty} P(a'<\pi_n<b')$ határérték, akkor ez valamennyi π_0-ra létezik, és független a kezdeti π_0-tól. Ezt az állítást a következő tételből vezethetjük le.

3.1. TÉTEL. A $\{\xi_k\}$ folyamat egy adott realizációjára, valamint egy $\pi_0=x$ és egy $\pi_0=y$ kezdeti értékre tekintsük a (2.3) és (2.3') transzformációk által generált $\pi_{n,x}$ és $\pi_{n,y}$ sorozatot. Ekkor van olyan C konstans, hogy majdnem minden realizációra

$$|\pi_{n,x}-\pi_{n,y}|<e^{-C\cdot n+o(n)}.$$

Bizonyítás. Ha a π_n és $\sigma_n := 1-\pi_n$ valószínűségeket 2 dimenziós vektorként fogjuk fel, akkor a (2.3) és (2.3′) képletek a következő

$$\begin{pmatrix}\pi_n\\ \sigma_n\end{pmatrix} = \lambda_1 \begin{pmatrix}1-P & P\\ P & 1-P\end{pmatrix}\begin{pmatrix}1-p_0 & 0\\ 0 & 1-p_1\end{pmatrix}\begin{pmatrix}\pi_{n-1}\\ \sigma_{n-1}\end{pmatrix}, \quad \text{ha} \quad \xi_n = 0, \tag{3.2}$$

$$\begin{pmatrix}\pi_n\\ \sigma_n\end{pmatrix} = \lambda_2 \begin{pmatrix}1-P & P\\ P & 1-P\end{pmatrix}\begin{pmatrix}p_0 & 0\\ 0 & p_1\end{pmatrix}\begin{pmatrix}\pi_{n-1}\\ \sigma_{n-1}\end{pmatrix}, \quad \text{ha} \quad \xi_n = 1 \tag{3.2′}$$

képletekkel helyettesíthetők, ahol λ_1, λ_2 alkalmas normáló tényezők. Tehát adott $\xi_1, \ldots, \xi_n$ relációhoz létezik olyan Λ normáló tényező, hogy

$$\begin{pmatrix}\pi_n\\ \sigma_n\end{pmatrix} = \Lambda \cdot \mathbf{A}(\xi_1, \ldots, \xi_n)\begin{pmatrix}\pi_0\\ \sigma_0\end{pmatrix},$$

ahol $\mathbf{A}(\xi_1, \ldots, \xi_n)$ egy $2n$ tényezős mátrix szorzat, amelyben a (3.2) és (3.2′) képletekben szereplő

$$\begin{pmatrix}1-P & P\\ P & 1-P\end{pmatrix}\begin{pmatrix}1-p_0 & 0\\ 0 & 1-p_1\end{pmatrix}, \quad \text{ill.} \quad \begin{pmatrix}1-P & P\\ P & 1-P\end{pmatrix}\begin{pmatrix}p_0 & 0\\ 0 & p_1\end{pmatrix}$$

szorzatok a realizációnak megfelelően váltakoznak. Egy Λ' normáló tényező bevezetésével elérhetjük azt, hogy a szorzat valamennyi tényezőjének a determinánsa 1 legyen. Λ' is pozitív, hiszen $P<1/2$ esetén a szorzat tényezőinek determinánsa pozitív. (A $P \geqq 1/2$ eset gyakorlati szempontból érdektelen.)

Legyen

$$\bar{A}(n) := \begin{pmatrix}\bar{a}_{11}(n) & \bar{a}_{12}(n)\\ \bar{a}_{21}(n) & \bar{a}_{22}(n)\end{pmatrix} := \frac{\Lambda}{\Lambda'} A(\xi_1, \ldots, \xi_n).$$

Ha az $\bar{A}(n)$ mátrix legalább három eleme ∞-hez tart ($n \to \infty$ esetén), akkor a tétel állítása a következőképpen igazolható: Elegendő belátni, hogy

$$\left|\frac{\pi_{n,x}}{\sigma_{n,x}} - \frac{\pi_{n,y}}{\sigma_{n,y}}\right| \to 0$$

mert π_n, σ_n határozottan el van választva a 0-tól, mivel π_n az $[a, b]$ intervallumra koncentrálódik (l. 1. ábra). Viszont

$$\frac{\pi_{n,x}}{\sigma_{n,x}} - \frac{\pi_{n,y}}{\sigma_{n,y}} = \frac{\bar{a}_{11}(n)x + \bar{a}_{12}(n)(1-x)}{\bar{a}_{21}(n)x + \bar{a}_{22}(n)(1-x)} - \frac{\bar{a}_{11}(n)y + \bar{a}_{12}(n)(1-y)}{\bar{a}_{21}(n)y + \bar{a}_{22}(n)(1-y)} \to 0$$

ui. $\bar{a}_{ij}(n) \xrightarrow[n\to\infty]{} \infty$ $i, j = 1, 2$ és $\det \bar{A}(n) = 1$. Végül belátjuk, hogy majdnem minden $\xi_1, \ldots, \xi_n$ realizációra $\bar{A}(n)$ minden eleme ∞-hez tart exponenciálisan. Mivel valamennyi tényező elemei nem negatív számok $\bar{a}_{11}(n) > b_{11}(n)$, ahol $b_{11}(n)$ a tényezők bal felső elemeinek szorzata; és $\bar{a}_{22}(n) > b_{22}(n)$, ahol $b_{22}(n)$ a tényezők jobb alsó elemeinek szorzata. Legyen n_0, ill. n_1 a $\xi_1, \ldots, \xi_n$ realizációban előforduló nullák,

ill. egyesek száma ($n_0+n_1=n$). Ekkor

$$b_{11}(n) = \left(\frac{1-p_0}{\sqrt{(1-p_0)(1-p_1)}}\right)^{n_0}\left(\frac{p_0}{\sqrt{p_0p_1}}\right)^{n_1}\left(\frac{1-P}{\sqrt{1-2P}}\right)^{n},$$

$$b_{22}(n) = \left(\frac{1-p_1}{\sqrt{(1-p_0)(1-p_1)}}\right)^{n_0}\left(\frac{p_1}{\sqrt{p_0p_1}}\right)^{n_1}\left(\frac{1-P}{\sqrt{1-2P}}\right)^{n},$$

$$\text{(3.3)}\quad \frac{1}{n}\log b_{11}(n) = \log\left(\frac{1-P}{\sqrt{1-2P}}\right)+\frac{n_0}{n}\log\left(\frac{1-p_0}{\sqrt{(1-p_0)(1-p_1)}}\right)+\frac{n_1}{n}\log\left(\frac{p_0}{\sqrt{p_0p_1}}\right).$$

$$\text{(3.3')}\quad \frac{1}{n}\log b_{22}(n) = \log\left(\frac{1-P}{\sqrt{1-2P}}\right)+\frac{n_0}{n}\log\left(\frac{1-p_1}{\sqrt{(1-p_0)(1-p_1)}}\right)+\frac{n_1}{n}\log\left(\frac{p_1}{\sqrt{p_0p_1}}\right).$$

Mivel a $\{\xi_k\}$ folyamat keverő, a (3.3) és a (3.3') kifejezések határértéke egy valószínűséggel egy-egy konstanssal egyenlő. Mivel

$$\log\left(\frac{1-p_1}{\sqrt{(1-p_0)(1-p_1)}}\right) = -\log\left(\frac{1-p_0}{\sqrt{(1-p_0)(1-p_1)}}\right),$$

illetve

$$\log\left(\frac{p_0}{\sqrt{p_0p_1}}\right) = -\log\left(\frac{p_1}{\sqrt{p_0p_1}}\right),$$

azért

$$\text{(3.4)}\quad \frac{1}{n}(\log b_{11}(n)+\log b_{22}(n)) = 2\log\left(\frac{1-P}{\sqrt{1-2P}}\right) > 0.$$

A (3.4) egyenlőtlenségből következik, hogy a $\lim_{n\to\infty}\frac{1}{n}\log b_{11}(n)$ és $\lim_{n\to\infty}\frac{1}{n}\log b_{22}(n)$ határértékek közül legalább az egyik pozitív. Ez utóbbi állításból következik, hogy $\lim_{n\to\infty} b_{11}(n)=\infty$, $\lim_{n\to\infty} b_{22}(n)=\infty$ relációk közül legalább az egyik teljesül. Mivel az

$$\begin{pmatrix}1-P & P\\ P & 1-P\end{pmatrix}\begin{pmatrix}1-p_0 & 0\\ 0 & 1-p_1\end{pmatrix} \quad \text{ill.} \quad \begin{pmatrix}1-P & P\\ P & 1-P\end{pmatrix}\begin{pmatrix}p_0 & 0\\ 0 & p_1\end{pmatrix}$$

szorzat mátrixok csak határozottan pozitív elemekből állnak, ezért akár $b_{11}(n)\to\infty$, akár $b_{22}(n)\to\infty$, fennállnak az $\bar{a}_{12}(n)\to\infty$, $\bar{a}_{21}(n)\to\infty$ relációk is.

KÖVETKEZMÉNY. A folytonos[1] sűrűségfüggvényű eloszlások körében a (3.1') egyenlet megoldása egyértelmű. Az egzisztencia kérdése még nyitott, az intervallum transzformációkra vonatkozó legujabb eredmények (l. pl. [5]) alapján negatív válasz várható.

[1] Csak az $[a, b]$ $(a=f_0(a), b=f_1(b))$ intervallumon követeljük meg a folytonosságot.

4. Döntési eljárás

A továbbiakban megfogalmazunk egy, a $\{\xi_k\}$ folyamattal kapcsolatos döntési feladatot.

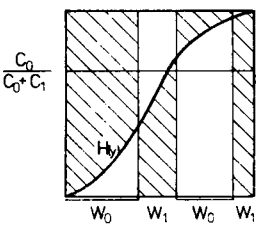
A $\xi_1, \ldots, \xi_n$ realizáció alapján el akarjuk dönteni, hogy az η_n vezérlő folyamat értéke 0-val, vagy 1-gyel egyenlő-e. A (2.1. lemma alapján ehhez a π_n értékére van szükségünk. Legyen c_0 annak a költsége, hogy $\eta_n=0$-t döntünk és $\eta_n=1$; c_1 pedig annak a költsége, hogy $\eta_n=1$-et döntünk és $\eta_n=0$. Döntési eljárásunkat a stacionárius (n-től független) stratégiákra korlátozzuk. Jelölje $W_0 \subset [0, 1]$ azt a tartományt, amelyre $\pi_n \in W_0$ esetén $\eta_n=0$-t döntünk, és legyen $W_1=[0, 1] \setminus W_0$. A π_n a posteriori valószínűség definíciójából következik, hogy a W_0 kritikus tartományhoz tartozó döntési eljárás várható költsége

$$c_0 \int_{W_0} (1-H(y))\,dy + c_1 \int_{W_1} H(y)\,dy, \tag{4.1}$$

ahol $H(y)$ a $\{\pi_n\}$ folyamat stacionárius eloszlásfüggvénye.

(A 2. ábrán a bevonalkázott tartomány területét adja az integrálok összege.)

Könnyen igazolható, hogyha W_0 tartomány véges sok intervallum egyesítése és $H(y)$ folytonos, akkor a (4.1) kifejezés $W_0=\left\{y: y>H^{-1}\left(\frac{c_0}{c_0+c_1}\right)\right\}$ tartományon minimális. A fenti döntés szemléletes értelme a következő: $\eta_n=0$-t akkor döntünk, ha π_n nagyobb, mint a $H(y)$ eloszlás $\frac{c_0}{c_0+c_1}$ kvantilise. Bizonyításképp megjegyezzük a következőt: ha W_0-nak egy teljes részintervalluma a kvantilistől balra van, akkor a költség csökkenthető azáltal, hogy ezt az intervallumot W_1-hez csatoljuk; fordított állítás érvényes W_1-nek a kvantilistől jobbra eső részintervallumaira is.



2. ábra

Megjegyzés. Szimmetrikus stacionárius eloszlás esetén hasonló eljárással optimalizálható a döntési eljárás abban az általánosabb esetben, ha a „váltások" átlagos gyakorisága is költségként jelentkezik.

5. Numerikus eredmények

A rekurzív egyenletet interpoláció nélkül oldottuk meg. A 200 iteráció utáni eredményeket az 1a, 1b táblázatok tartalmazzák. A π_n a posteriori valószínűség stacionárius sűrűségfüggvényét a [0, 1] intervallum 100 ekvidisztans osztópontjában adjuk meg. Az 1b táblázat a függvényértékeket logaritmikus skálán adja meg. A grafikus ábrázolás a 3a, 3b ábrákon található. Folytonos vonal jelöli az egyenlet megoldását.

Szimuláltuk is az eredeti folyamatot. A 200 000 lépés után kialakult empirikus sűrűségfüggvény értékeit, az egyenlet megoldásához hasonlóan a 2a, 2b táblázatok tartalmazzák.

1a. TÁBLÁZAT

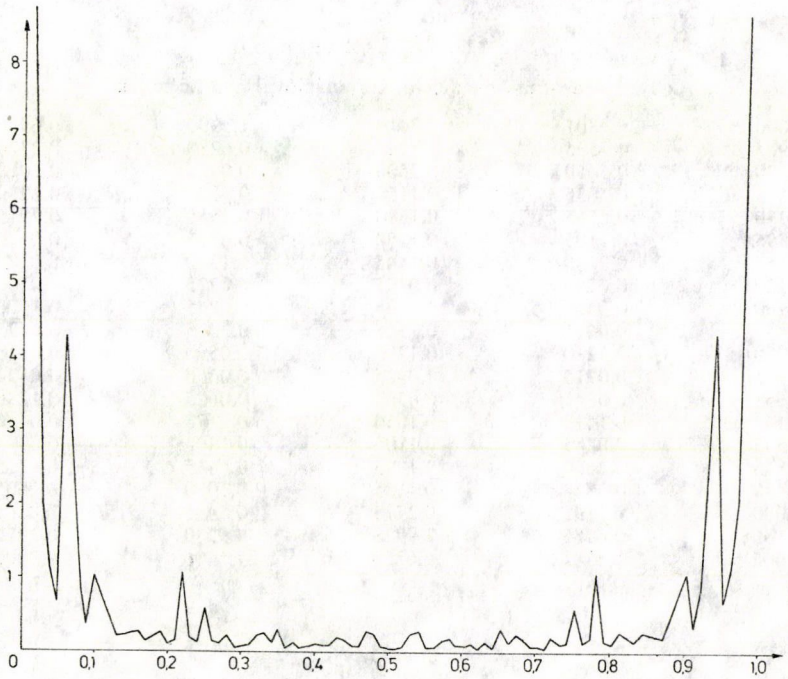
A π_n a posteriori valószínűség stacionárius sűrűségfüggvényének értékei a [0, 1] intervallum 100 ekvidisztans osztópontjában

21,9350	6,8126	2,0783	1,1670	0,6782
4,3132	2,4030	0,8538	0,3544	1,0412
0,7697	0,4837	0,2177	0,2314	0,2390
0,2664	0,1645	0,2036	0,2607	0,1008
0,1615	1,0728	0,1996	0,1204	0,5835
0,1457	0,1185	0,2075	0,0584	0,0854
0,0931	0,2078	0,2483	0,1211	0,2923
0,0602	0,1343	0,0498	0,0623	0,1099
0,0789	0,0909	0,1823	0,1541	0,0731
0,0741	0,2751	0,2420	0,0756	0,0480
0,0480	0,0756	0,2420	0,2751	0,0741
0,0731	0,1541	0,1824	0,0909	0,0789
0,1099	0,0623	0,0498	0,1343	0,0601
0,2923	0,1211	0,2483	0,2078	0,0931
0,0854	0,0584	0,2075	0,1185	0,1457
0,5835	0,1204	0,1996	1,0727	0,1615
0,1008	0,2607	0,2036	0,1645	0,2664
0,2390	0,2313	0,2177	0,4837	0,7697
1,0412	0,3544	0,8537	2,4030	4,3133
0,6782	1,1670	2,0783	6,8124	21,9345

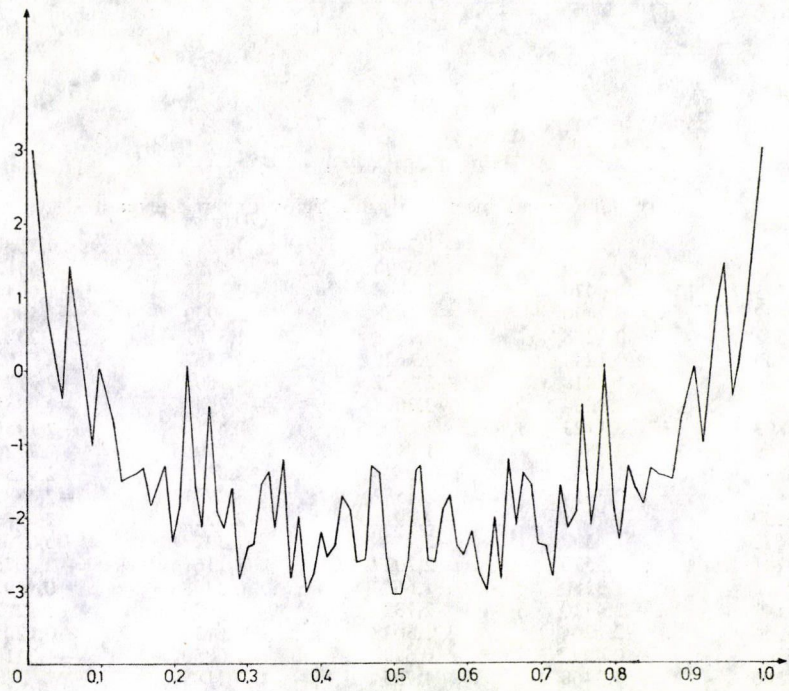
1b. TÁBLÁZAT

A π_n a posteriori valószínűség stacionárius sűrűségfüggvénye logaritmusának értékei a [0, 1] intervallum 100 ekvidisztans osztópontjában

3,0881	1,9188	0,7316	0,1545	−0,3883
1,4617	0,8767	−0,1581	−1,0373	0,0404
−0,2618	−0,7263	−1,5244	−1,4638	−1,4311
−1,3226	−1,8051	−1,5915	−1,3444	−2,2948
−1,8231	0,0703	−1,6115	−2,1169	−0,5388
−1,9265	−2,1332	−1,5728	−2,8401	−2,4607
−2,3736	−1,5711	−1,3933	−2,1112	−1,2301
−2,6109	−2,0078	−3,0006	−2,7765	−2,2085
−2,5400	−2,3980	−1,7019	−1,8703	−2,6165
−2,6020	−1,2907	−1,4187	−2,5828	−3,0360
−3,0360	−2,5828	−1,4186	−1,2907	−2,6020
−2,6165	−1,8702	−1,7018	−2,3980	−2,5400
−2,2085	−2,7765	−3,0006	−2,0078	−2,8109
−1,2300	−2,1112	−1,3933	−1,5710	−2,3736
−2,4608	−2,8401	−1,5728	−2,1332	−1,9265
−0,5388	−2,1169	−1,6115	0,0702	−1,8232
−2,2948	−1,3444	−1,5915	−1,8051	−1,3226
−1,4311	−1,4638	−1,5244	−0,7263	−0,2618
0,0404	−1,0373	−0,1582	0,8767	1,4617
−0,3884	0,1544	0,7316	1,9187	3,0881



3a. ábra



3b. ábra

2a. TÁBLÁZAT

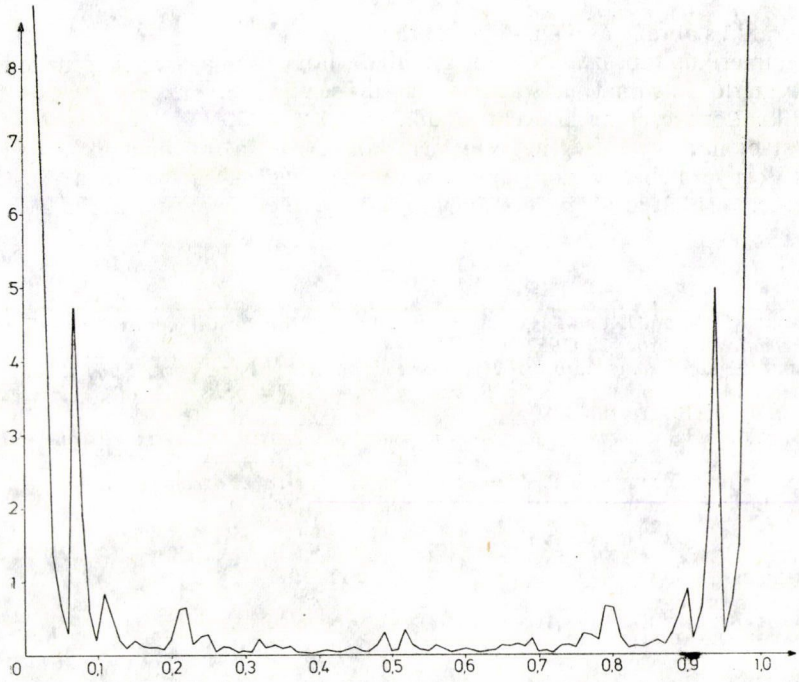
Az 1a. táblázatnak megfelelő, szimulációval nyert értékek

0,0000	26,4801	5,7410	1,3900	0,6930
0,2905	4,7535	1,8945	0,6280	0,2120
0,8570	0,5450	0,2500	0,0955	0,2010
0,1515	0,1055	0,1150	0,0825	0,2230
0,6330	0,6565	0,1680	0,2655	0,2915
0,0895	0,1195	0,1150	0,0355	0,0590
0,0455	0,2140	0,1145	0,1380	0,0985
0,1120	0,0550	0,0405	0,0225	0,0505
0,0750	0,0445	0,0455	0,0845	0,1170
0,0490	0,0615	0,1365	0,3075	0,0625
0,0790	0,3210	0,1405	0,0590	0,0510
0,1175	0,0915	0,0425	0,0460	0,0830
0,0655	0,0245	0,0555	0,0625	0,1230
0,1145	0,1410	0,1050	0,2175	0,0460
0,0535	0,0285	0,1105	0,1205	0,0995
0,2845	0,2650	0,1895	0,6435	0,6255
0,2430	0,0845	0,1185	0,1010	0,1605
0,2090	0,1195	0,2715	0,5415	0,8865
0,2165	0,6255	1,9730	4,9780	0,2920
0,7005	1,4145	6,0220	27,6196	0,0000

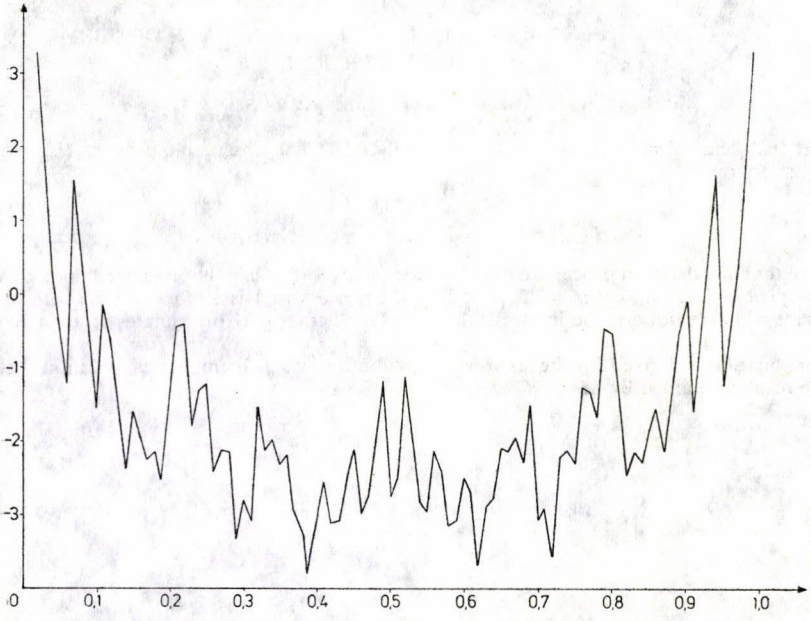
2b. TÁBLÁZAT

Az 1b. táblázatnak megfelelő, szimulációval nyert értékek

	3,2764	1,7476	0,3293	−0,3667
−1,2361	1,5589	0,6390	−0,4652	−1,5512
−0,1543	−0,6070	−1,3863	−2,3486	−1,6044
−1,8872	−2,2490	−2,1628	−2,4950	−1,5006
−0,4573	−0,4208	−1,7838	−1,3261	−1,2327
−2,4135	−2,1244	−2,1628	−3,3382	−2,8302
−3,0900	−1,5418	−2,1672	−1,9805	−2,3177
−2,1893	−2,9004	−3,2064	−3,7942	−2,9858
−2,5603	−3,1123	−3,0900	−2,4710	−2,1456
−3,0159	−2,7887	−1,9914	−1,1793	−2,7726
−2,5383	−1,1363	−1,9625	−2,8302	−2,9759
−2,1413	−2,3914	−3,1582	−3,0791	−2,4889
−2,7257	−3,7091	−2,8914	−2,7726	−2,0956
−2,1672	−1,9590	−2,2538	−1,5256	−3,0791
−2,9281	−3,5578	−2,2027	−2,1161	−2,3076
−1,2570	−1,3280	−1,6634	−0,4408	−0,4692
−1,4147	−2,4710	−2,1328	−2,2926	−1,8295
−1,5654	−2,1244	−1,3038	−0,6134	−0,1205
−1,5302	−0,4692	0,6796	1,6050	−1,2310
−0,3560	0,3468	1,7954	3,3185	



4a. ábra



4b. ábra

A grafikus ábrázolás a 4a, 4b ábrákra került.

A numerikus tapasztalatok azt mutatják, hogy P nagy értékei mellett (ritkán vált a vezérlő folyamat) a stacionárius sűrűségfüggvény az $a=f_0(a)$ és $b=f_1(b)$ pontok kis környezeteire koncentrálódik.

A stacionárius sűrűségfüggvény nem konvex, ui. a folyamat az $a=f_0(a)$ pont egy kis környezetében viszonylag nagy valószínűséggel van és innen pozitív valószínűséggel kerül át az $f_1(a)$ pont egy kis környezetébe, és így tovább.

IRODALOM

[1] GIRSHIK, M. A. and RUBIN, H., "A *Bayes approach* to a quality control model", *Annals of Mathematical Statistics* **23** (1952) 114—125.

[2] LEHMAN, E. L., *Testing Statistical Hypothesis* (John Wiley, New York, 1957).

[3] RUDEMO, M., "Doubly stochastic *Poisson processes* and process control", *Advanced Applied Probability* **4** (1972) 318—338.

[4] Ширяев, А. Н., *Статистический последовательный анализ* (Наука, Москва, 1976) 243—259.

[5] COLLET, P. and ECKMANN, J-P., *Iterated maps on the interval as dynamical systems* (Birkhauser, Basel—Boston—Stuttgart, 1980).

(Beérkezett: 1979. szeptember 20.)

KRÁMLI ANDRÁS ÉS LUKÁCS PÁL
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1111 BUDAPEST, KENDE U. 13—17.

VASSEL RÓBERT
SZÁMÍTÓGÉPALKALMAZÁSI KUTATÓ INTÉZET
1536 BUDAPEST, PF. 227.

A DECISION PROBLEM RELATED TO A DISCRETE DOUBLY STOCHASTIC PROCESS

A. KRÁMLI, P. LUKÁCS and R. VASSEL

Let $\{\eta_n\}$ be a *Markov chain* with two states 0 and 1. The observable process $\{\xi_n\}$ is governed by $\{\eta_n\}$ as follows:

$$P(\xi_n = 1|\eta_n = 0,\ \eta_{n-1}, \ldots, \xi_{n-1}, \ldots) = p_0,$$

$$P(\xi_n = 1|\eta_n = 1,\ \eta_{n-1}, \ldots, \xi_{n-1}, \ldots) = p_1.$$

We prove that any decision procedure based on the past of $\{\xi_n\}$ can be expressed in terms of the posterior probabilities $\pi_n = P(\eta_n = 1|\xi_n, \ldots, \xi_0)$. The process $\{\pi_n\}$ is a *Markov chain* with continuous state space which is not ergodic in classical sense. The theorem 3.1. formulates an ergodic property of $\{\pi_n\}$.

An equation is derived for the stationary distribution of π_n. Its numerical solution is compared with the results obtained by *Monte-Carlo method.*

Alkalmazott Matematikai Lapok 6 (1980) 105—122

LINEÁRIS PROGRAMOZÁS RÉSZBEN RENDEZETT VEKTORTEREKBEN (A. G. Pinszker eredményei)

B. NAGY ANDRÁS

Budapest

Ebben a tanulmányban A. G. PINSZKER, leningrádi matematikus eredményeit ismertetjük, amelyek a lineáris programozás (LP) elméletét általánosítják oly módon, hogy az LP feladat némely paraméterének értékéül nem valós számot, hanem egy absztrakt, részben rendezett vektortér elemét választják. Ez a tér lehet valós szám n-esek etere, valamely intervallumon értelmezett függvények (folytonosak, szakaszosan folytonosak, *Riemann szerint integrálhatók, Lebesgue szerint integrálhatók* stb.) tere, lineáris operátorok tere stb.

Az LP eddig ismert általánosításaitól a legfontosabb eltérés, hogy a célfüggvény értéke sem valós szám, hanem absztrakt elem.

A tanulmányban ismertetendő feladatszerkezet valószínűleg jól alkalmazható olyan dinamikus gazdasági feladatok matematikai modellezéséhez, ahol az erőforrások kapacitása, valamint a termelés értékelése az időben változik. Ezeknél a feladatoknál a részben rendezett tér szerepét az időintervallumon értelmezett szakaszosan folytonos függvények tere játssza. Más interpretációban a feladat ugyan statikus, de a termelés párhuzamosan egyszerre több dimenzióban folyik, ezért a termelési folyamatokat nem számok, hanem szám n-esek írják le, ahol esetleg az n igen nagy is lehet.

Abban az egyszerűbb esetben, amikor az LP feladat strukturális paraméterei egy külső paraméter lineáris függvényei, a parametrikus programozás jól ismert esete áll elő.

A dolgozat végén függelék ismerteti a részben rendezett vektorterek elméletének egyes alapelemeit.

1. Bevezetés

Bevezetésként bemutatunk két tipikus feladatot, amelyekről intuitíve érezzük, hogy az LP-hez közelálló problémákról van szó, de amíg az első példa esetében az LP hagyományos módszerei még célra vezethetnek, ha feleslegesen sok munkával is, addig a második példa esetében ezek a módszerek közvetlenül már egyáltalán nem alkalmazhatók.

1. FELADAT. Egy kereskedelmi hálózat m boltból és n szállító üzemből áll, p különböző árucikk forgalmazásával foglakozik. Az i-edik bolt a kereslet statisztikai felmérése alapján a k-adik áruból a_{ik} mennyiséget rendel ($i=1, n$; $k=1, \ldots, p$). A j-edik üzem a k-adik áruból maximálisan b_{jk} mennyiséget tud szállítani ($j=1, \ldots, m$). A k-adik áru j-edik üzemből az i-edik boltba c_{ijk} pénzegységért szállítható. Olyan szállítási tervet kell kidolgozni, hogy valamennyi áru a legkisebb szállítási költséggel jusson el a boltokba. Vezessük be a következő p-elemű vektorokat:

$$\mathbf{a}_i = [a_{ik}], \quad \mathbf{b}_j = [b_{jk}], \quad \mathbf{c}_{ij} = [c_{ijk}] \quad \text{és} \quad \mathbf{x}_{ij} = [x_{ijk}], \quad \mathbf{c}_{ij} \circ \mathbf{x}_{ij} = [c_{ijk} x_{ijk}],$$

ahol x_{ij} az egyelőre ismeretlen kontingens (i—j viszonylatban) p-elemű vektorát jelöli.

Ezekkel a jelölésekkel a feladat matematikai alakja:

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{c}_{ij}\circ\mathbf{x}_{ij}\to\min \tag{1.1}$$

$$\sum_{i=1}^{m}\mathbf{x}_{ij}=\mathbf{b}_j \quad (j=1,\ldots,n)$$

$$\sum_{j=1}^{n}\mathbf{x}_{ij}=\mathbf{a}_i \quad (i=1,\ldots,m)$$

$$\mathbf{x}_{ij}\geqq\mathbf{0} \quad (i=1,\ldots,m\ j=1,\ldots,n).$$

Itt a célfüggvény értéke nem szám, hanem egy p-elemű vektor. Az optimális megoldás célfüggvény értékének minden komponense nem kisebb, mint bármely megengedett megoldás célfüggvény értékének megfelelő komponense.

Látjuk, hogy az (1.1.) feladat megoldása ekvivalens p darab közönséges szállítási feladat megoldásával. Ezzel kapcsolatban a következő kifogásaink lehetnek:

— ha p egy nagyobb természetes szám, akkor p darab azonos típusú szállítási feladat egyenkénti megoldása nagyon munkaigényes lehet;

— az LP érzékenység vizsgálatainak elmélete alapján gyanítható, hogy a p darab feladat optimális kötött elem szerkezetei néhány (p-nél általában jóval kevesebb) osztályba sorolhatók, ahol az egy osztályba tartozó optimális megoldások eléréséhez minden feladaton ugyanazokat a lépéseket kell elvégezni. Tehát utólag kiderül, hogy milyen sok felesleges számítást végeztünk.

2. Feladat. Egy zöldségtermesztő mezőgazdasági üzem n különféle primőr árut termeszt. A primőrszezonban (T) az árak az idő függvényében alakulnak. A j-edik primőr értékesítési ára (sokéves tapasztalat alapján) a $c_j(t)$ függvényt követi. A termelés feltételei egyenlőtlenségekkel fejezhetők ki, ahol az egyenlőtlenség szabad tagjai (a kapacitások) szintén az idő függvényei. Az i-edik kapacitás változását a $b_i(t)$ függvény írja le. A termesztés technológiája (az egyenlőtlenségek koefficiensei) egy konstans mátrixszal jellemezhető: $[a_{ij}]$.

Keressük a termelési tervet $([x_j(t)])$, amely a primőrszezon minden időpotjában a maximális árbevételt biztosítja.

A feladat matematikai alakja:

$$\sum_{j=1}^{m}c_j(t)\cdot x_j(t)\to\max \tag{1.2}$$

$$\sum_{j=1}^{m}a_{ij}x_j(t)\leqq b_i(t) \quad (i=1,\ldots,n)$$

$$x_j(t)\geqq 0 \qquad (j=1,\ldots,m)$$

$$t\in T.$$

Jóllehet ez a feladat is feltűnően emlékeztet a hagyományos LP feladatokra, itt még elvileg sem lehet a megoldást közvetlenül visszavezetni hagyományos feladatok megoldására, mivel a pontonkénti értelmezés kontinuum számosságú feladat megoldását tételezi fel.

A két feladatban minden esetre található közös vonás. Mindkét esetben számok helyett egy lineáris tér elemei szerepeltek. Az első esetben p-elemű vektorok, a második esetben a T intervallumon értelmezett szakaszosan folytonos függvények.

Az első térben a lineáris müveleteket és a parciális rendezést komponensenként, a másodikban pontonként értelmeztük.

A. G. PINSZKER azt vizsgálta, hogy nem lehetne-e ezen az alapon általánosítani az LP elméletét és meglepően egyszerű megoldásokat talált ([6], [7], [8], [9]).

A lineáris programozásnak sokféle általánosítását ismerjük. Ezekben általában a megengedett halmaz egy absztrakt lineáris tér (topologikus vektortér, *Banach-tér, Hilbert-tér* stb.) konvex részhalmaza, a célfüggvény pedig egy lineáris funkcionál, amely a megengedett halmaz elemeit a valós számegyenesre képezi le. PINSZKER általánosítása ettől lényegesen eltér, mivel itt nem valós szám a célfüggvény értéke, hanem absztrakt elem.

PINSZKER általánosításához legközelebb a parametrikus programozás áll, azonban a parametrikus programozás csak lineáris paraméter-függvények esetén rendelkezik kielégítő algoritmusokkal. A *Pinszker-féle általánosítás* alkalmas nemlineáris paraméter-függvények kezelésére is, sőt sokkal általánosabb problémák megoldására is.

Dolgozatunk matematikai apparátusának lényeges eleme a részben rendezett *(Riesz-* vagy *Kantorovics-)terek* elmélete. A funkcionálanalízis ezen fontos ága Magyarországon kevéssé ismert, jóllehet gyökerei RIESZ FRIGYES munkásságához nyúlnak vissza ([10]). Ezért a dolgozat végén ismertetjük ennek az elméletnek az alapelemeit, VULICH B. Z. monográfiája alapján (oroszul [2], angolul [3]). A dolgozat érdemi részében ezeket az alapelemeket ismertnek tételeztük fel.

2. Az LP feladat általánosításai

Mint ismeretes, az LP alapfeladatának kanonikus alakja a következő:

$$(2.1)\quad \max_{\{\lambda_j\}} \left\{ \sum_{j=1}^{n} \gamma_j \lambda_j \,\middle|\, \sum_{j=1}^{n} \alpha_{ij} \lambda_j = \beta_i \geqq 0 \ (i = 1, \ldots, m);\ \lambda_j \geqq 0 \ (j = 1, \ldots, n) \right\}.$$

A feladat a következő paraméterekből épül fel:

γ_j — a célfüggvény együtthatói;
β_i — a korlátozó feltételek szabad tagjai;
α_{ij} — a korlátozó feltételek együtthatói;
λ_j — a feladat meghatározandó ismeretlenjei, változói.

$i = 1, 2, \ldots, m$ a korlátozó feltétel sorszáma, $j = 1, 2, \ldots, n$ az ismeretlen sorszáma.

Többféle általánosítási lehetőség adódik. A. G. PINSZKER a következő alapváltozatokat vizsgálta:

A feladat

A korlátozó feltételek szabad tagjai és az ismeretlenek valamely *arkhimédészi K-lineár* elemei;

B feladat

A célfüggvény együtthatói valamely *arkhimédészi K-lineár* elemei;

C feladat

A célfüggvény együtthatói, a korlátozó feltételek szabad tagjai és az ismeretlenek olyan *arkhimédészi K-lineár* elemei, mely *kvázigyűrű* is;

D feladat

Minden paraméter az előbbihez hasonló *arkhimédészi kvázigyűrű* eleme.

Ebben a tanulmányban csak az A, B és C feladatokkal foglalkozunk.

Az LP (2.1) alakú kanonikus feladatából az általánosítások a következő helyettesítésekkel nyerhetők: [1]

A feladat

$$\beta_i \to b_i \in \mathfrak{X}^+, \quad \lambda_j \to x_j \in \mathfrak{X}^+;$$

B feladat

$$\gamma_j \to c_j \in \mathfrak{X};$$

C feladat

$$\beta_i \to b_i \in \mathfrak{X}^+, \quad \lambda_j \to x_j \in \mathfrak{X}^+, \quad \gamma_j \to c_j \in \mathfrak{X}.$$

Az egyszerűbb tárgyalásmód kedvéért feltesszük, hogy mindhárom változat esetén ugyanabból a kanonikus LP feladatból indultunk ki, azaz az azonos betűk ténylegesen azonos objektumokat jelentenek, bármely feladatról is legyen szó. Ez a feltételezés, ami nem kötelez semmire, lehetővé teszi, hogy ne ismételgessünk gondolatmeneteket.

Meg kell jegyezni, hogy a B feladat nem más, mint a többcélfüggvényes vektorprogramozás általánosítása, ezért megoldhatósága sokkal kétségesebb, mint az A és C feladatoké.

A három feladat legfontosabb közös tulajdonsága, hogy a célfüggvények a megengedett halmazt az $\mathfrak{X}$ K-lineárba képezik le, ezért az optimális megoldást szigorúbban kell definiálni, mint a közönséges LP esetén.

DEFINÍCIÓ. Az A, C, illetve B feladatok optimális megoldása olyan $\mathbf{X}^0 = \{x_j^0\}$, illetve $\Lambda^0 = \{\lambda_j^0\}$ vektor, mely megengedett (azaz kielégíti a korlátozó feltételeket) és a hozzá tartozó célfüggvény érték összehasonlítható a többi megengedett megoldáshoz tartozó célfüggvény értékkel, azok maximuma, azaz velük egyenlő, vagy náluk nagyobb.

Nyilván itt nem mondható egyszerűen, hogy „nem kisebb", hiszen ami nem kisebb az lehet nem összehasonlítható is. A probléma éles volta éppen abban a kérdésben jelentkezik: nem túl erős megkötés-e olyan megoldást keresni, mely minden más megoldással összehasonlítható? Mint ismeretes, hasonló probléma merül fel

[1] Itt és a továbbiakban a következő jelölési elvet alkalmazzuk:
— a latin ábécé első (a, b, c ...) és utolsó (...x, y, v, w, z) kisbetűi mindig az $\mathfrak{X}$ részben rendezett tér elemeit jelölik;
— θ az $\mathfrak{X}$ részben rendezett tér nulla eleme;
— egyéb görög kisbetűk valós vagy természetes (index) számokat jelölnek;
— indexként szokásosan a latin ábécé középső kisbetűi (...i, j, k, l, m, n, ..., s, t, r ...) szerepelnek;
— minden más jel halmazt, függvényt vagy más matematikai objektumot jelöl.

a vektorprogramozás esetében is, és ott általában nincs kielégítő megoldás. Mint látni fogjuk, az A és C feladatok esetében a helyzet sokkal kedvezőbb. A B feladat, mint már említettük, sajnos örökli a vektorprogramozás problémáit.

Vizsgálatainkban a *szimplex módszert,* illetve a *Gauss-féle bázistranszformációt* alkalmazzuk, ezért figyelmünket a feladatok bázismegoldásaira (úgy a megengedettekre, mint a nem megengedettekre — az előjelkorlátozást nem kielégítőekre) kell fordítanunk.

Az alábbiakban egy darabig mindegy lesz, melyik feladatról van szó a három közül.

A feladat bármely bázistranszformációjában szereplő $\mathfrak{X}$-ből való elemek a kiindulási alak absztrakt elemeinek lineáris kombinációi. Ezeket a kombinációkat A. G. PINSZKER *megoldó kombinációknak* nevezi. Jelölje M az összes megoldó kombináció halmazát: $M \subseteq \mathfrak{X}$. Ez az illető feladat *megoldó halmaza.* Tekintsük az $u = \sum_{x \in M} |x| \in X^+$ elemet. Ha $u = \theta$, akkor $M = \{\theta\}$ és a feladat egyetlen és optimális megoldása $\mathbf{X}^0 = \{\theta, \theta, \ldots \theta\}$, illetve (B feladat esetén) $A^0 = \{0, 0, \ldots 0\}$.

Tekintsünk el ettől a triviális esettől, legyen $u \neq \theta$. Akkor az $\mathfrak{X}$ K-lineár $\mathfrak{X}_u$ főkomponense, melyet u feszít ki, tartalmazza a megoldó halmazt, annak valamennyi lineáris kombinációját, tehát a teljes feladatot is, abban az értelemben, hogy a feladat minden megoldásának komponensei (A és C feladatok esetén) és minden előforduló célfüggvényérték az $\mathfrak{X}_u$ elemei. Mivel $\mathfrak{X}$ *arkhimédészi K-lineár,* $\mathfrak{X}_u$ szintén az. Tehát beágyazható, mondjuk, egy Y K_σ-térbe. Tekintsük most az Y tér u által kifeszített főkomponensének Y_u' korlátos elemek terét. Nyilván Y_u' szintén „tartalmazza" a feladatot. Ilyen módon elegendő azt az esetet vizsgálni, amikor a feladat absztrakt tere $\mathfrak{X}$ egy unitér K_σ-tér korlátos elemeinek tere. Minden más eset visszavezethető ugyanis erre.

A későbbiekben szükséges lesz a következő két lemma.

1. LEMMA. Legyen E az $\mathfrak{X}$ K-lineár véges részhalmaza. Ekkor $\mathfrak{X}$ felbontható véges számú komponens részlineárra úgy, hogy E bármely elemének vetülete a felbontás bármely komponens részlineárjára összehasonlítható a θ elemmel.

Bizonyítás. Legyen $E = \{x_1, x_2, \ldots x_s\}$. Alkalmazzunk teljes indukciót. Az $\mathfrak{X}$ felbontható $\{\mathfrak{X}_{x_1^+}, \mathfrak{X}_{x_1^+}^d\}$ főkomponensekre. Ezekre E minden eleme vetíthető és x_1 vetületei összehasonlíthatók θ-val. Tegyük fel, hogy az $\{\mathfrak{X}_1, \mathfrak{X}_2, \ldots \mathfrak{X}_r\}$ felbontás, legalábbis x_s kivételével, E minden elemére kielégíti a lemma feltételét. Akkor az $\{\mathfrak{X}_{1x_s^+}, \mathfrak{X}_{1x_s^+}^d, \ldots, \mathfrak{X}_{rx_s^+}, \mathfrak{X}_{rx_s^+}^d\}$ felbontás már E minden elemére kielégíti ezt a feltételt, tehát az általunk keresett felbontás, mely nem több, mint 2^s elemű. Az indukció elve alapján a lemma igaz.

2. LEMMA. Legyen $\mathfrak{X}$ egy K_σ-tér korlátos elemeinek tere, E olyan véges részhalmaza, melynek elemei összehasonlíthatók θ-val. Ekkor létezik olyan $f: \mathfrak{X} \to R$ pozitív additív funkcionál, melyre

$$f(x) \begin{cases} > 0, & \text{ha} \quad x \geqq 0, \ x \neq 0 \\ = 0, & \text{ha} \quad x = 0 \\ < 0, & \text{ha} \quad x \leqq 0, \ x \neq 0 \end{cases} \quad (x \in E).$$

Bizonyítás. Alkalmazzuk a *Hahn—Banach-tételt.* E szerint minden $x \in E$ elemhez hozzárendelhető egy f_x pozitív additív funkcionál, ahol

$$f_x(|x|) = \|x\|.$$

Könnyen belátható, hogy az $f = \sum_{x \in E} f_x$ funkcionál éppen az általunk keresett tulajdonságú pozitív additív funkcionál, azaz a lemma igaz.

Meg kell jegyezni, hogy az első lemma bizonyítása konstruktív, azaz algoritmust ad a felbontás tényleges megkonstruálásához. A második lemma bizonyítása viszont jellegezetes egzisztencia bizonyítás. Szerencsére a gyakorlatban előfordulható feladatok többségénél a kívánt funkcionál adott valamilyen integrál formájában.

Most pedig térjünk vissza a feladatokhoz. Két esetet kell megkülönböztetnünk: a megoldó halmaz elemei összehasonlíthatók θ-val, illetve nem hasonlíthatók össze. Az első esetben mindhárom feladat egységesen kezelhető. A második esetben, mivel M véges halmaz, alkalmazható az 1. lemma, és $\mathfrak{X}$ felbontható, mondjuk r számú főkomponensre. Ekkor az A és C feladatokból r számú vetületfeladat képezhető úgy, hogy minden absztrakt elem helyére a főkomponenseken vett vetületét tesszük. Könnyen belátható a vetítés lineáris voltából, hogy

a) ha az eredeti feladat megengedett halmaza nem üres, akkor a vetületfeladatoké sem az;

b) ha az eredeti feladat célfüggvénye korlátos a megengedett halmazon, akkor a vetületfeladatok célfüggvényei is így viselkednek;

c) ha valamennyi vetületfeladatnak van optimális megoldása, akkor ezek összege az eredeti feladat optimális megoldását adja.

A B feladat korlátozó feltételeiben nincs absztrakt elem, ezért ennek a feladatnak nem képezhető vetületfeladata. Azonban a felbontás mintegy sávokra bontja az absztrakt célfüggvényt, ezért mint látni fogjuk, a B feladat az 1. lemma alkalmazásával egy egyszerűbb vektorprogramozási feladatra vezethető vissza.

Az A és C feladatok esetében tehát a vetületfeladatok képzésével a második eset visszavezethető az első esetre. A továbbiakban csak az első esetet kell vizsgálnunk.

Következő lépésként megvizsgáljuk a megoldó halmaz konkrét szerkezetét mindhárom feladat esetében. Ennél a vizsgálatnál mindig tartjuk magunkat ahhoz a kényelmi kikötéshez, hogy az azonos jelek azonos objektumokat jelentenek mindhárom feladatban. Ez vonatkozik elsősorban az $[\alpha_{ij}]_{m \times n}$ mátrixra. Erről a mátrixról feltételezzük, hogy sorai lineárisan függetlenek. Ellenkező esetben a függő sorok, illetve a hozzájuk tartozó egyenletek a korlátozó feltételek közül kizárhatók.

Tehát $m \leqq n$ és az $[\alpha_{ij}]$ mátrixból kiválasztható s ($1 \leqq s \leqq C_n^m$) nem szinguláris m-edrendű négyzetes mátrix, azaz a feladatoknak s számú (nem feltétlenül különböző és nem feltétlenül megengedett) bázis megoldásuk van. Ezekhez s-számú *Gauss-féle bázistranszformáció* tartozik.

Tekintsük most az A feladatot, illetve annak k-adik bázismegoldását. Jelöljük ennek bázis inverzét $[\bar{\alpha}_{ij}^k]_{m \times n}$ szimbólummal. Ekkor a korlátozó feltételek k-adik bázistranszformációja a következő alakot ölti:

$$x_{j_k} = \sum_{i=1}^{m} \bar{\alpha}_{ij_k}^k \Big(b_i - \sum_{j_k' \in I_{sz}^k} \alpha_{ij_k'} x_{j_k'}\Big) \quad (j_k \in I_B^k), \tag{2.2}$$

ahol I_B^k a k-adik bázistranszformáció bázis indexeinek, I_{sz}^k a szabad változók indexeinek halmaza. Mivel a bázismegoldásban $x_j^k=\theta$, ha $j\in I_{sz}^k$, ezért az A feladat megoldó halmaza

$$M_A=\left\{x_{j_k}^k \mid k=1,2,\ldots,s;\ x_{j_k}^k=\sum_{i=1}^{m}\bar{\alpha}_{ij_k}^k b_i\ \ (j_k\in I_B^k)\right\}\cup\{\theta\}. \tag{2.3}$$

A B feladat (2.2) típusú kifejezésében nincs absztrakt elem:

$$\lambda_{j_k}=\sum_{i=1}^{m}\bar{\alpha}_{ij_k}^k\Big(\beta_i-\sum_{j_k'\in I_{sz}^k}\alpha_{ij_k'}\lambda_{j_k'}\Big)\quad (j_k\in I_B^k). \tag{2.4}$$

Rendezzük át ezt a kifejezést és vezessünk be néhány rövidítést

$$\lambda_{j_k}=\pi_{j_k}-\sum_{j_k'\in I_{sz}^k}\omega_{j_kj_k'}\lambda_{j_k'}\quad (j_k\in I_B^k), \tag{2.5}$$

ahol $\pi_{j_k}=\sum_{i=1}^{m}\bar{\alpha}_{ij_k}^k\beta_i$, $\omega_{j_kj_k'}=\sum_{i=1}^{m}\bar{\alpha}_{ij_k}^k\alpha_{ij_k'}$ valós számok. A bázisváltozók ilyen alakú kifejezését helyettesítsük be a célfüggvénybe:

$$\begin{aligned}\sum_{j=1}^{n}c_j\lambda_j &= \sum_{j_k\in I_B^k}c_{j_k}\Big(\pi_{j_k}-\sum_{j_k'\in I_{sz}^k}\omega_{j_kj_k'}\lambda_{j_k'}\Big)+\sum_{j_k'\in I_{sz}^k}c_{j_k'}\lambda_j=\\ &=\sum_{j_k\in I_B^k}c_{j_k}\pi_{j_k}+\sum_{j_k'\in I_{sz}^k}\Big(c_{j_k'}-\sum_{j_k\in I_B^k}\omega_{j_kj_k'}c_{j_k}\Big)\lambda_{j_k'}=\\ &=\sum_{j_k\in I_B^k}c_{j_k}\pi_{j_k}+\sum_{j_k'\in I_{sz}^k}c_{j_k'}^k\lambda_{j_k'}\to\max.\end{aligned} \tag{2.6}$$

Itt $c_{j_k'}^k=c_{j_k'}-\sum_{j_k\in I_k'}\omega_{j_kj_k'}c_{j_k}$. Tehát a B feladat megoldó halmaz:

$$M_B=\{x_{j_k}^k \mid k=1,2,\ldots,s;\ x_{j_k}^k=c_{j_k}\ (j_k\in I_B^k);\ x_{j_k}^k=c_{j_k}^k\ (j_k\in I_{sz}^k)\}. \tag{2.7}$$

A C feladat (2.2) típusú kifejezése megegyezik az A feladatéval, (2.6) típusú kifejezése pedig a következő

$$\sum_{j=1}^{n}c_jx_j=\sum_{j_k\in I_B^k}c_{j_k}x_{j_k}^k+\sum_{j_k'\in I_{sz}^k}c_{j_k'}^kx_{j_k'}\to\max. \tag{2.8}$$

Itt $c_{j_k}, c_{j_k'}^k\in M_B$, $x_{j_k}^k$, $x_{j_k'}\in M_A$. Következésképpen, mint az várható is volt, $M_C=M_A\cup M_B$. Ennek az összefüggésnek az alapján a tanulmány fő tételét csak a C feladatra bizonyítjuk, de egyszerű megfontolásokkal belátható, hogy az igaz az A és B feladatokra is.

TÉTEL. Ha a feladat megengedett halmaza nem üres, és ezen a célfüggvény felülről korlátos, valamint a megoldó halmaz minden eleme összehasonlítható θ-val, akkor a feladatnak van optimális megoldása.

Bizonyítás. A második lemmában definiált pozitív additív funkcionál segítségével hozzá rendelünk a feladathoz egy közönséges numerikus LP feladatot:

C feladat	*C′ feladat*
$\sum_{j=1}^{n} c_j x_j \to \max$	$\sum_{j=1}^{n} f(c_j)\lambda_j \to \max$
$\sum_{j=1}^{n} \alpha_{ij} x_j = b_i \quad (i = 1, 2, \ldots, m)$	$\sum_{j=1}^{n} \alpha_{ij}\lambda_j = f(b_i) \quad (i = 1, 2, \ldots, m)$
$x_j \geqq \theta \quad (j = 1, 2, \ldots, n)$	$\lambda_j \geqq \theta \quad (j = 1, 2, \ldots, n).$

Belátható, hogy ha az eredeti feladat megengedett halmaza nem üres, akkor a hozzárendelt feladaté sem az. Ezt a legegyszerűbben úgy ellenőrizhetjük, ha a szimplex-algoritmust párhuzamosan alkalmazzuk mindkét feladatra. Ez az eljárás kölcsönösen egyértelmű megfeleltetést teremt a két feladat bázismegoldásai között. Ezeket a továbbiakban röviden megfeleltetett megoldásoknak fogjuk nevezni.

Megmutatjuk, hogy ha az eredeti feladat célfüggvénye korlátos a megengedett halmazon, akkor ilyen a hozzárendelt feladaté is.

Tegyük fel az ellenkezőjét, azaz hogy a C′ feladat célfüggvénye felülről nem korlátos a megengedett halmazon. Akkor, mint az az LP elméletéből ismert, van olyan bázistranszformáció (célszerüen átindexelve):

$$\gamma'_0 + \sum_{j=m+1}^{n} \gamma'_j \lambda_j \to \max \tag{2.9}$$

$$\lambda_i + \sum_{j=m+1}^{n} \alpha'_{ij}\lambda_j = \beta'_i \quad (i = 1, 2, \ldots, m)$$

$$\lambda_j \geqq 0 \qquad (j = 1, 2, \ldots, n),$$

ahol $\gamma'_{m+1} > 0$ és minden i-re $\alpha'_{i,m+1} \leqq 0$. Megfeleltetett lépésekkel a C feladat is hasonló alakra hozható, ahol γ'_j és β'_i helyett az M megoldó halmaz megfelelő c'_j $(f(c'_j)=\gamma'_j)$ és $b'_i(f(b'_i)=\beta'_i)$ elemei állnak. Mivel M elemei összehasonlíthatók θ-val, ezért $c_{m+1} \geqq \theta$ és $c_{m+1} \neq \theta$, ellenkező esetben ugyanis $c'_{m+1} \leqq \theta$ lenne, amiből $\gamma'_{m+1} = f(c'_{m+1}) \leqq 0$ következne. Ekkor viszont minden δ-ra ($\delta > 0$ valós szám) az

$$X^{\delta} = \{b'_1 - \delta\alpha'_{1,m+1}u, \ldots, b'_m - \delta\alpha'_{m,m+1}u, \delta u, \theta, \ldots, \theta\} \tag{2.10}$$

a C feladat megengedett megoldása. Ha itt δ minden határon túlnő, akkor a $c'_0 + c'_{m+1}\delta u$ célfüggvényérték is korlátlanul nő, ami ellentmond eredeti feltételezésünknek. Tehát a C′ feladat célfüggvénye is korlátos kell, hogy legyen.

Ha viszont a C′ feladat megengedett halmaza nem üres, célfüggvénye ezen a halmazon felülről korlátos, akkor az LP egzisztencia tétele alapján van optimális megoldása. Nem nehéz belátni, hogy a C feladat megfeleltetett bázismegoldása szintén optimális. Valóban, a C′ feladat optimális bázismegoldásához is tartozik egy (2.9.) alakú transzformáció, csakhogy itt minden $\gamma'_j \leqq 0$ $(j = m+1, \ldots n)$. A C feladat megfeleltetett transzformációjában a célfüggvény

$$c'_0 + \sum_{j=m+1}^{u} c'_j x_j \to \max \tag{2.11}$$

alakú. Itt, nyilván $c'_j \leqq 0$ minden $j=m+1, \ldots n$ esetén (ez f tulajdonságából következik). Legyen a C' feladat optimális megoldása A^0, a neki megfeleltetett C feladatbeli megoldás

$$X^0 = \{x_1^0, x_2^0, \ldots, x_m^0, \theta, \ldots, \theta\}.$$

Legyen továbbá $X=\{x_j\}$ a C feladat egy tetszőleges megoldása. Ekkor

$$\sum_{j=1}^{n} c_j x_j^0 = c'_0 + \sum_{j=m+1}^{n} c'_j \theta = c'_0 \geqq c'_0 + \sum_{j=m+1}^{n} c'_j x_j = \sum_{j=1}^{n} c_j x_j, \tag{2.12}$$

tehát X^0 valóban optimális megoldás.

Ezzel a tételt bebizonyítottuk.

Az 1. lemma és a tétel az A és C feladatok számára algoritmussal szolgál. Összefoglaljuk ezt az algoritmust, anélkül, hogy számítástechnikai részletekre is kitérnénk.

a) a feladat adataiból meghatározzuk a megoldó halmazt;

b) ha a megoldó halmaz elemei nem összehasonlíthatók θ-val, az 1. lemma alapján $\mathfrak{X}$-et komponens alterekre bontjuk és előállítjuk a vetületfeladatokat;

c) megfelelő pozitív additív funkcionál segítségével (általában integrálással) kiszámítjuk a hozzárendelt LP feladat(ok) paramétereit, majd megoldjuk ez(eke)t. Figyelmünket nem a számszerű eredményekre, hanem a szimplex (vagy más megoldó) algoritmus elvégzett lépéseire összpontosítjuk.

d) A hozzárendelt feladat(ok) optimumhoz vezető algoritmusát megismételjük az eredeti (vetület) feladto(ko)n is. Ha a b) lépés kimaradt, akkor ez az optimális eredményt adja, ellenkező esetben a vetületoptimumok összege adja a kiindulási feladat optimális megoldását. Ha valamelyik vetületfeladat nem megoldható, akkor az eredeti feladat nem elégíti ki a tétel feltételeit és így, természetesen, nem megoldható.

A B feladattal kapcsolatban, ha a fenti algoritmus *b)* pontja kihagyható, akkor az algoritmus alkalmazható. Ha M_B elemei nem hasonlíthatók össze θ-val, akkor az 1. lemma alkalmazásával nem vetületfeladatokat kaphatunk, hanem egy olyan absztrakt vektorprogramozási feladatot, melyhez a 2. lemma funkcionáljával egy közönséges numerikus vektorprogramozási feladat rendelhető. Ebben az esetben a vektorprogramozás valamely módszerét alkalmazva *Pareto-optimumot* vagy kompromisszumos megoldást kereshetünk, melynek megfeleltethető az eredeti feladat megoldása. Ez hasonló tulajdonságú lesz, ami a fent alkalmazott gondolatmenettel minden esetben igazolható.

3. Numerikus példa

Illusztrációképpen bemutatjuk egy A típusú feladat megoldását. Ez a feladat szállításifeladat lesz, mivel a szállítási feladatok esetében a megoldó halmaz szerkezete jelentősen leegyszerűsödik.

Az A típusú absztrakt szállítási feladat a következő:

$$(3.1) \qquad \min \sum_{i=1}^{m} \sum_{j=1}^{n} \gamma_{ij} x_{ij}$$

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

$$x_{ij} \in \mathfrak{X}^+.$$

Bizonyítható, hogy a megoldó halmaz szerepét a következő halmaz játszhatja: $M_A = \{x^k \mid x^k = \sum_{i=1}^{m} \delta_i^k a_i - \sum_{j=1}^{n} \delta_j^k b_j;\ \delta_l^k = 0 \text{ vagy } 1;\ l = 1, 2, \ldots m,\ m+1, \ldots m+n\}$.

A megoldó halmaz szerepéből következik, hogy további egyszerűsítésként M_A-ból kizárható az az elem, mely (-1)-szerese egy másik M_A-beli elemnek. Ez egyébként vonatkozik az általános esetre is.

Tekintsük tehát a következő szállítási feladatot: $\mathfrak{X}$ a [0, 3] szakaszon értelmezett *Riemann szerint integrálható* függvények tere, amit $C^R_{[0,3]}$ jellel fogunk jelölni;

$$[\gamma_{ij}] = \begin{bmatrix} 4 & 3 & 1 & 0 \\ 1 & 0 & 4 & 2 \\ 0 & 3 & 2 & 5 \end{bmatrix}, \quad \begin{array}{ll} a_1 = 2t, & b_1 = t^3, \\ a_2 = t^2, & b_2 = b_4 = t, \\ a_3 = t^3, & b_3 = t^2, \end{array} \quad t \in [0, 3].$$

A feladat megoldó halmaza 21 elemből áll:

$$(3.2)\ M_A = \{2t, t^2, t^3, 2t+t^2, 2t+t^3, t^2+t^3, 2t+t^2+t^3, t, t^3+t, t^2+t, 2t-t^3, 2t-t^2,$$
$$2t+t^2-t^3, 2t+t^3-t^2, t^2-t, t^2-t^3, t^2+t^3-t, t^3-t^2-t, t^2-t^3-t,$$
$$t+t^2+t^3\}.$$

Az 1. lemmának megfelelően X-et a következő főkomponensekre bontjuk:

$$\mathfrak{X}_1 = C^R_{[0,\alpha_1)}, \quad \mathfrak{X}_2 = C^R_{[\alpha_1,1)}, \quad \mathfrak{X}_3 = C^R_{[1,\sqrt{2})},$$

$$\mathfrak{X}_4 = C^R_{[\sqrt{2},\alpha_2)}, \quad \mathfrak{X}_5 = C^R_{[\alpha_2,2)}, \quad \mathfrak{X}_6 = C^R_{[2,3)},$$

ahol

$$\alpha_1 = \frac{-1+\sqrt{5}}{2}, \quad \alpha_2 = \frac{1+\sqrt{5}}{2}.$$

Az α_2 a $t^3 - t^2 - t$ polinom pozitív gyöke a [0, 3] intervallumban. A számítások elvégzésekor kiderült, hogy ez az M_A-beli elem sehol sem fordul elő, ezért az $\mathfrak{X}_4$ és $\mathfrak{X}_5$ főkomponensek helyett az $\mathfrak{X}_{45} = C^R_{[\sqrt{2},2)}$ főkomponens szerepeltethető. A $C^R_{[\alpha,\beta]}$, illetve $C^R_{[\alpha,\beta)}$ alakú főkomponensekben a 2. lemma szerinti funkcionál szerepét az

$$(3.3) \qquad \int_{\alpha}^{\beta} \cdot\, dt$$

alakú *Riemann-integrál* játssza.

Az első főkomponensre vetített feladat hozzárendelt feladatának optimális táblázata:

	0,036	0,191	0,079	0,191
0,382	4	0,112 ③	0,079 ①	0,191 ⓪
0,075	0,000 ①	0,079 ⓪	4	2
0,036	0,036 ⓪	3	2	5

A vetületfeladat megoldása:

$$\mathrm{pr}_{\bar{x}_1}[x_{ij}^0] = \begin{bmatrix} \theta & t-t^2 & t^2 & t \\ \theta & t^2 & \theta & \theta \\ t^3 & \theta & \theta & \theta \end{bmatrix}_{t\in[0,\alpha_1)}.$$

A második komponens altérre vetített feladat hozzárendelt optimális táblázata:

	0,214	0,309	0,255	0,309
0,618	4	0,054 ③	0,255 ①	0,309 ⓪
0,255	0,000 ①	0,255 ⓪	4	2
0,214	0,214 ⓪	3	2	5

A vetület megfeleltetett megoldása:

$$\mathrm{pr}_{\bar{x}_2}[x_{ij}^0] = \begin{bmatrix} \theta & t-t^2 & t^2 & t \\ \theta & t^2 & \theta & \theta \\ t^3 & \theta & \theta & \theta \end{bmatrix}_{t\in[\alpha_1,1)}.$$

A harmadik főkomponensre vetített feladat hozzárendelt optimális táblázata:

	0,750	0,500	0,609	0,500
1,000	4	3	0,609 ①	0,391 ⓪
0,609	0,000 ①	0,500 ⓪	4	0,109 ②
0,750	0,750 ⓪	3	2	5

A megfeleltetett megoldás:

$$\mathrm{pr}_{\mathfrak{X}_3}[x_{ij}^0] = \begin{bmatrix} \theta & \theta & t^2 & 2t-t^2 \\ \theta & t & \theta & t^2-t \\ t^2 & \theta & \theta & \theta \end{bmatrix}_{t\in[1,\sqrt{2})}.$$

Az összevont negyedik és ötödik főkomponensre vetített feladat hozzárendelt optimális táblázata:

	3,000	1,000	1,724	1,000
2,000	4	3	1,724 ①	0,276 ⓪
1,724	0,000 ①	1,000 ⓪	4	0,724 ②
3,000	3,000 ⓪	3	2	5

A segítségével nyert megfeleltetett megoldás:

$$\mathrm{pr}_{\mathfrak{X}_{45}}[x_{ij}^0] = \begin{bmatrix} \theta & \theta & t^2 & 2t-t^2 \\ \theta & t & \theta & t^2-t \\ t^3 & \theta & \theta & \theta \end{bmatrix}_{t\in[\sqrt{2},2)}.$$

Végül az utolsó, hatodik főkomponens vetület feladatához hozzárendelt optimális táblázat:

	16,250	2,500	6,333	2,500
5,000	4	3	2,500 ①	2,500 ⓪
6,333	3,833 ①	2,500 ⓪	4	2
16,250	12,417 ⓪	3	3,833 ②	5

A megfeleltetett optimális megoldás:

$$\mathrm{pr}_{\mathfrak{X}_6}[x_{ij}^0] = \begin{bmatrix} \theta & \theta & t & t \\ t^2-t & t & \theta & \theta \\ t^3+t-t^2 & \theta & t^2-t & \theta \end{bmatrix}_{t\in[2,3]}.$$

Az eredeti feladat optimális megoldása tehát:

$$x_{11}^0 = \theta$$

$$x_{12}^0 = \begin{cases} t-t^2, & t\in[0,1) \\ \theta, & t\in[1,3] \end{cases}$$

$$x_{13}^0 = \begin{cases} t^2, & t\in[0,2) \\ t, & t\in[2,3] \end{cases}$$

$$x_{14}^0 = \begin{cases} t, & t\in[0,1) \\ 2t-t^2, & t\in[1,2) \\ t, & t\in[2,3] \end{cases}$$

$$x_{21}^0 = \begin{cases} \theta, & t\in[0,2) \\ t^2-t, & t\in[2,3] \end{cases}$$

$$x_{22}^0 = \begin{cases} t^2, & t\in[0,1) \\ t, & t\in[1,3] \end{cases}$$

$$x_{23}^0 = \theta$$

$$x_{24}^0 = \begin{cases} \theta, & t\in[0,1) \\ t^2-t, & t\in[1,2) \\ \theta, & t\in[2,3] \end{cases}$$

$$x_{31}^0 = \begin{cases} t^3, & t\in[0,2) \\ t^3-t^2+t, & t\in[2,3] \end{cases}$$

$$x_{32}^0 = \theta$$

$$x_{33}^0 = \begin{cases} 0, & t\in[0,2) \\ t^2-t, & t\in[2,3] \end{cases}$$

$$x_{34}^0 = \theta$$

A célfüggvény optimális értéke:

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\gamma_{ij}x_{ij}^0 = \begin{cases} 3t-2t^2, & t\in[0,1) \\ 3t^2-2t, & t\in[1,3] \end{cases}$$

4. Függelék.

A részben rendezett vektorterek elméletének alapelemei

Legyen $\mathfrak{X}$ tetszőleges halmaz valamely rendezési relációval, amelyet szokásosan $\geqq$ jellel jelölünk.

Azt mondjuk, hogy $\mathfrak{X}$ rendezési struktúra vagy *háló,* ha minden véges részhalmazának létezik legkisebb felső határa (supremum) és legnagyobb alsó határa (in-

fimum). A következő jelöléseket fogjuk alkalmazni: legyen $E=\{x_1, \dots x_n\}$, akkor E suprémuma $\sup E=x_1 \vee x_2 \dots \vee x_n$, E infimuma $\inf E=x_1 \wedge x_2 \dots \wedge x_n$.

A $\vee$ és $\wedge$ műveletek egyenként asszociatívak és kommutatívak, ezenkívül kölcsönösen disztributívak. Hálóelméleti szempontból tehát a rendezési háló disztributív.

Azt mondjuk, hogy az $\mathfrak{X}$ rendezési háló *σ-teljes (teljes)*, ha valamennyi korlátos és megszámlálható (tetszőleges számosságú) részhalmazának létezik supremuma és infimuma.

Az $\mathfrak{X}$ hálót *lineáris hálónak* vagy *K-lineárnak* (L. V. KANTOROVICS, az elmélet első rendszeres kutatója nevének kezdő betűje alapján) nevezzük, ha benne lineáris struktúra van értelmezve és ez a rendezési struktúrával a következő összefüggésben van:

K.1. Ha $x, y, z \in \mathfrak{X}$ és $x \geqq y$, akkor $x+z \geqq y+z$,

K.2. Ha $x, y \in \mathfrak{X}, \alpha \geqq 0$ és $x \geqq y$, akkor $\alpha x \geqq \alpha y$.

Jellegzetes *K-lineár* a számegyenes valamely intervallumán értelmezett integrálható függvények tere, amelyben a lineáris műveleteket és a parciális rendezést „természetes módon", pontonként értelmezzük.

Nem minden K.1. és K.2. feltételt kielégítő részben rendezett vektortér *K-lineár*. Ehhez az is kell, hogy az illető tér rendezési struktúrája háló legyen. Viszont tetszőleges részben rendezett vektortérben értelmezhető a *pozitív elemek* fogalma.

Mindazokat az $x \in \mathfrak{X}$ elemket, melyekre $x \geqq \theta$ (θ a tér nulla eleme) az $\mathfrak{X}$ tér pozitív elemeinek nevezzük és halmazukat az $\mathfrak{X}^+$ jellel jelöljük. Definíció szerint $\theta \in X^+$. Értelemszerűen az $\mathfrak{X}^- = (-1) \cdot \mathfrak{X}^+$ halmaz a negatív elemek halmaza, mely nem azonos a nem pozitív elemek halmazával. Mindkét említett halmaz kihegyezett konvex kónusz, melyek egyetlen közös pontja a θ.

A fentebb említett függvénytér pozitív elemei a minden pontban nem negatív függvények.

Minden *K-lineárban* érvényesek a következő azonosságok:

$$x+\sup E = \sup (x+E) \quad (x \in \mathfrak{X}, E \subseteq \mathfrak{X}) \tag{4.1}$$

$$\alpha \sup E = \sup (\alpha E), \text{ ha } \alpha \geqq 0; = \inf (\alpha E), \text{ ha } \alpha < 0 \quad (E \subseteq \mathfrak{X}) \tag{4.2}$$

$$x+y = x \wedge y + x \vee y \quad (x, y \in \mathfrak{X}). \tag{4.3}$$

Legyen tehát X *K-lineár*. Ekkor minden elemre értelmezhető

— az $x \in \mathfrak{X}$ elem pozitív része: $x^+ = x \vee \theta$,

— az $x \in \mathfrak{X}$ elem negatív része: $x^- = (-x)^+ = (-x) \vee \theta$,

— az $x \in \mathfrak{X}$ elem abszolút értéke: $|x| = x^+ + x^-$.

Nyilván bármely $x \in \mathfrak{X}$ esetén $x^+, x^-, |x| \in \mathfrak{X}^+$. (4.3) alapján

$$x = x^+ - x^-. \tag{4.4}$$

Bizonyítható, hogy x^+ és x^- a legkisebb pozitív elemek, amelyekkel a (4.4) típusú felbontás felírható.

Illusztratív példánkban $f^+(t)=f(t)$, ha $f(t)\geqq 0$ és $f^+(t)=0$, ha $f(t)<0$, $f^-(t)=0$, ha $f(t)\geqq 0$ és $f^-(t)=-f(t)$, ha $f(t)<0$, valamint $|f|(t)=|f(t)|$.

Az $\mathfrak{X}$ *K-lineár* x és y elemeit *diszjunktaknak* nevezzük, ha $|x|\wedge|y|=\theta$. Az E és F részhalmazok akkor diszjunktak, ha *Descartes-szorzatuk* $(E\times F)$ csupa diszjunkt párból áll. Jelölések: xdy, EdF. Ha $E=\{x\}$, akkor egyszerűen xdF írható.

Bizonyítható, hogy x^+dx^- és a (4.4) típusú felbontást ez az egyetlen pozitív diszjunkt pár elégíti ki.

Ha EdF, akkor e részhalmazok lineáris burkai is diszjunktak, és $E\cap F\subseteqq\{\theta\}$.

Példánkban két függvény akkor és csak akkor diszjunkt, ha egyazon pontban mindig legalább az egyik értéke nulla.

Valamely $\mathfrak{X}$ K-lineárt *egységelemesnek, unitérnek* mondjuk, ha van olyan $u\in\mathfrak{X}^+$ eleme, amely csak a nullával diszjunkt. Természetesen ez az u elem nem egyértelmű, mivel K.1. és K.2. miatt megfelelő u esetén $\alpha u+x$ ($\alpha>0$, $x\in\mathfrak{X}^+$) szintén megfelelő lesz. Sok esetben azonban célszerű valamely megfelelő u elemet rögzíteni, mint az $\mathfrak{X}$ unitér tér *egységelemét*. A mi példánk választott terünk unitér és egységeleméül az $u(t)\equiv 1$ függvényt szokás választani.

Az $e\in\mathfrak{X}$ elemet *egységdarabnak* nevezzük, ha $e\wedge(u-e)=\theta$, ahol u az unitér tér egységeleme. Nyilván, ha e egységdarab, akkor $\theta\leqq e\leqq u$. Egy rögzített u egységelemhez tartozó egységdarabok az illető tér rendezési bázisát alkotják, melynek természetesen elemei u és θ is. E bázis a két rendezési művelettel ($\vee$ és $\wedge$), valamint az $\bar{e}=u-e$ komplementer képzéssel *Boole-algebra*. Az így származtatott *Boole-algebrák* vizsgálata fontos kutatási terület.

Ha az $\mathfrak{X}$ *K-lineár* rendezési struktúrája σ-teljes, akkor *K_σ-térnek*, ha pedig teljes, akkor *K-térnek* nevezzük. A nemzetközi szakirodalom használja még a *Riesz-tér*, illetve a *teljes vektorháló* elnevezéseket is.

Példaként választott terünk *K-tér*.

A mi céljaink szempontjából a *K-* és *K_σ-terek* közötti különbség nem játszik szerepet, mivel legerősebb megállapításaink is csak *K_σ-terekre* vonatkoztak. Egyébként minden *K-tér* egyben *K_σ-tér* is.

Fontos azonban a *K-lineárok* és a *K_σ-terek* közötti különbség.

A valós számok természetes rendezéssel *K-lineárt* alkotnak (ami egyben *K-tér* is). Ebben a *K-lineárban* érvényes az úgynevezett *Arkhimédész-elv*, mely tetszőleges részben rendezett vektortérre a következőképpen mondható ki:

ha valamely $x\in\mathfrak{X}^+$ elemre az $\{nx\}$ $n=1, 2\ldots$ sorozat felülről korlátos, akkor $x=0$

Mindazokat a *K-lineárokat*, amelyek kielégítik ezt az elvet, arkhimédészinek nevezzük. Minden *K_σ-tér arkhimédészi K-lineár*, de konstruálható *nem arkhimédészi K-lineár* is. Például a lexikografikusan rendezett szám n-esek tere nem arkhimédészi, ami azért meglepő, mert hiszen ez a rendezés nem is parciális, hanem lineáris.

Azt mondjuk, hogy az $\mathfrak{X}$ részben rendezett halmaz *beágyazható* az $\mathfrak{Y}$ részben rendezett halmazba, ha létezik olyan rendezéstartó, kölcsönösen egyértelmű U leképezés (izomorfizmus) X és $\mathfrak{Y}'\subseteqq\mathfrak{Y}$ között, melyre $\sup E$ és $\inf E$ létezése esetén $\sup U(E)=U(\sup E)$ és $\inf U(E)=U(\inf E)$. Bizonyítható, hogy minden *arkhimédészi K-lineár* beágyazható valamely *K_σ-térbe*, ahol az illető *K-lineár* képe egyben lineáris altér is. Általában nincs akadálya annak, hogy a beágyazandó $\mathfrak{X}$ halmazt azonosítsuk beágyazott $\mathfrak{Y}'$ képével.

Legyen E az $\mathfrak{X}$ *K-lineár* valamely részhalmaza. Azt mondjuk, hogy E kielégíti a *normalitás feltételét,* illetve $\mathfrak{X}$ *normálisan tartalmazza E-t,* ha $x\in E$, $y\in\mathfrak{X}$ és $|x|\leqq|y|$ esetén $y\in E$. Ha $\mathfrak{X}_1$ az $\mathfrak{X}$ normálisan tartalmazott altere, akkor bizonyíthatóan maga is *K-lineár* és így *normális részlineárnak* nevezhető. Hasonlóan, ha $\mathfrak{X}$ K_σ- vagy *K-tér,* akkor az $\mathfrak{X}_1$ normális részlineár is ilyen típusú, ezért *normális altérnek* nevezhető.

Ha az $E\subseteq\mathfrak{X}$ részhalmaz tartalmazza valamennyi részhalmaznak $\mathfrak{X}$-ben létező suprémumait és infimumait, akkor *tökéletesnek* nevezzük. Ha $\mathfrak{X}_1$ az $\mathfrak{X}$ normális részlineárja és egyben tökéletes részhalmaza is, akkor az $\mathfrak{X}$ *komponens részlineárjának* nevezzük. Teljesen analóg módon definiálhatók a K_σ- és *K-terek komponens alterei* is. Az elnevezés érthetővé válik, ha például a valós szám n-esek terét tekintjük, mely a komponensenkénti rendezéssel *K-tér.* Ennek a *K-térnek* valamennyi komponens altere oly módon állítható elő, hogy kiválasztjuk azokat az elemeket, amelyekben *csak előre rögzített komponensek lehetnek nem nullák.* Egyes irodalmi források a szintén szemléletes *sáv* elnevezést használják, amit viszont a már korábban is vizsgált példánk, az integrálható függvények tere magyaráz meg. Ebben a térben a komponens alterek olyan alterek, amelyekben az elemek (függvények) csak *egy rögzített kompakt halmaz által meghatározott sávban, vagy sávokban vesznek fel nullától különböző értéket.*

Tekintsük az $\mathfrak{X}$ *K-lineár* azon legnagyobb részhalmazát, amely diszjunkt valamely $E\subseteq\mathfrak{X}$ részhalmazzal és nevezzük az E *diszjunkt kiegészítésének.* Jelben: E^d. Nyilván E^d az $\mathfrak{X}$ lineáris altere. Bizonyítható, hogy egyben komponens részlineár is. Az E^d diszjunkt kiegészítése E^{dd} igazolhatóan az a legkisebb komponens altér, amely még tartalmazza E-t, ezért az E *által kifeszített* komponens részlineárnak nevezzük. Könnyen belátható, hogy ha EdF, akkor $E^{dd}dF^{dd}$. Különösen fontos az az eset, amikor $E=\{x\}$. Ekkor $E^{dd}=\mathfrak{X}_x$ az x elem által kifeszített komponens részlineár. Minden $x\in\mathfrak{X}$ elem esetén $\mathfrak{X}_x$ unitér K-lineár, amiben egységelemként általában $|x|$-et rögzítik. Az elméletben fontos unitér komponens részlineárokat megkülönböztetésül *főkomponenseknek* nevezzük.

Legyen $\mathfrak{X}_1$ az $\mathfrak{X}$ K-lineár valamely komponens részlineárja. Egy $x\in\mathfrak{X}$ elem $\mathfrak{X}_1$-en vett *vetületének* az $y=\sup\{z|z\in\mathfrak{X}_1, z\leqq x\}$ elemet nevezzük és $\mathrm{pr}_{\mathfrak{X}_1}x$ jellel jelöljük (ha nem okoz félreértést, akkor az $\mathfrak{X}_1$ index el is hagyható: pr x). Az x elem vetületének létezése tetszőleges részlineárra nem garantálható, de bizonyítható, hogy tetszőleges *K-lineár* főkomponensére és *K-terek* tetszőleges komponens alterére minden pozitív elem vetíthető. Tetszőleges elem vetületét a pozitív rész és a negatív rész vetületeinek különbségeként definiáljuk (feltéve, hogy e vetületek léteznek): $\mathrm{pr}\,x=\mathrm{pr}\,x^+-\mathrm{pr}\,x^-$.

Bizonyítható, hogy $\mathrm{pr}_{\mathfrak{X}_1}x$ létezése maga után vonja $\mathrm{pr}_{\mathfrak{X}_1^d}x$ létezését és $x=\mathrm{pr}_{\mathfrak{X}_1}x+\mathrm{pr}_{\mathfrak{X}_1^d}x$, ami megmagyarázza a vetület elnevezést.

Hagyományos példánkban a komponens alteret egyértelműen meghatározza az alapintervallum valamely rögzített kompakt részhalmaza. Valamely elem (függvény) vetülete egy komponens altérre egy olyan függvény, mely az alteret meghatározó kompakt részhalmazon megegyezik a vetített függvénnyel, míg azon kívül mindenütt nulla.

A vetítés operátora lineáris, azaz $\mathrm{pr}\,(\alpha x+\beta y)=\alpha\,\mathrm{pr}\,x+\beta\,\mathrm{pr}\,y$ és rendezéstartó, azaz ha $x\geqq y$, akkor $\mathrm{pr}\,x\geqq\mathrm{pr}\,y$. $x\in\mathfrak{X}_1$ akkor és csak akkor, ha $\mathrm{pr}_{\mathfrak{X}_1}x=x$, és $x\in\mathfrak{X}_1^d$ akkor és csak akkor, ha $\mathrm{pr}_{\mathfrak{X}_1}x=\theta$.

Azt mondjuk, hogy az $\{\mathfrak{X}_i\}_{i\in I}$ komponens részlineár-család az $\mathfrak{X}$ *K-lineár felbontását* adja, ha

(4.5.) $\{\mathfrak{X}_i\}_{i\in I}$ elemei páronként diszjunktak;

(4.6.) 0 kivételével nincs $\mathfrak{X}$-ben olyan elem, amely $\{\mathfrak{X}_i\}_{i\in I}$ valamennyi elemével diszjunkt lenne;

(4.7.) $\mathfrak{X}$ minden eleme vetíthető $\{\mathfrak{X}_i\}_{i\in I}$ valamennyi elemére.

Az $\mathfrak{X}$ K-lineár jellegezetes felbontása $\{\mathfrak{X}_1, \mathfrak{X}_1^d\}$, ahol $\mathfrak{X}_1$ főkomponens $\mathfrak{X}$-ben.

Legyen $\mathfrak{X}$ unitér *K-lineár* u egységelemmel. Bizonyítható, hogy az $\mathfrak{X}' = \{x|\exists\lambda_x |x|\leqq\lambda_x u\}$ részhalmaz részlineár, azaz olyan lineáris altér, mely egyben maga is *K-lineár*. Ennek a fontos részlineárnak a neve az $\mathfrak{X}$ *korlátos elemeinek tere*. Mivel u az $\mathfrak{X}'$-ben is egységelem, ezért $\mathfrak{X}'$ maga is unitér. Ha $\mathfrak{X}$ K_σ- vagy K-tér, akkor $\mathfrak{X}'$ szintén ilyen típusú.

Az $\mathfrak{X}$ korlátos elemeinek $\mathfrak{X}'$ tere természetes módon normálható az $\|x\| = \inf\{\lambda_x| |x|\leqq\lambda_x u\}$ normával. Az ilyen módon konstruálható részben rendezett normált-, illetve *Banach-terek* vizsgálata az általános elmélet legfontosabb részeinek egyike. Mi itt mindössze az ezekre a terekre alkalmazott *Hahn—Banach-tételt* említjük meg, amire a pozitív funkcionálok bevezetése után fogunk visszatérni.

Az elmélet szempontjából döntő jelentőségű szerepe van a rendezési reláció alapján konstruálható konvergencia elvnek, illetve az általa indukált rendezési topológiának. Egy $\{x_n\}$ sorozat $\mathfrak{X}$-ben akkor *hálókonvergens* valamely x elemhez, azaz $x=(o)-\lim_{n\to\infty} x_n$, ha található olyan monoton növekvő $\{y_n\}$ sorozat és olyan monoton csökkenő $\{z_n\}$ sorozat, hogy $\sup y_n = x = \inf z_n$ és $y_n\leqq x_n\leqq z_n$ minden $n=1, 2 \ldots$ esetén. Megjegyezzük, hogy ez a konvergencia fogalom teszi a *K-tereket* a topologikus vektorterek harmadik nagy családjává a *Banach* és a *Hilbert-terek* mellett.

Legyen $\mathfrak{X}$ *K-lineár* és R a valós számegyenes. Az $f: \mathfrak{X}\to R$ leképezést *pozitív additív funkcionálnak* nevezzük, ha (0)-folytonos (azaz az (0)-konvergens sorozatokat konvergens számsorozatokba képezi le), additív (azaz $f(x+y)=f(x)+f(y)$) és minden $x\in\mathfrak{X}^+$ elemre $f(x)\geqq 0$. Bizonyítható, hogy a pozitív additív funkcionálok homogének is ((azaz $f(\alpha x)=\alpha f(x)$ minden α számra).

Most már megfogalmazható a *Hahn—Banach-tétel* minket érdeklő alakja:

Legyen $\mathfrak{X}$ valamely unitér *K-lineár* korlátos elemeinek tere. Akkor minden $x\in X^+$ elemhez található olyan f_x pozitív additív funkcionál, melyre $f_x(x)=\|x\|$. Ezt a tételt úgy is szokták megfogalmazni, hogy a korlátos elemek terén „elég sok pozitív additív funkcionál található".

Utolsónak vázlatosan érintjük a részben rendezett kommutatív gyűrű fogalmát. Az $\mathfrak{X}$ *K-lineárt* akkor nevezzük *kommutatív gyűrűnek*, ha benne szorzat értelmezhető a következő tulajdonságokkal:

$(x, y, z\in\mathfrak{X},\ \alpha\in R)$

G.1. $xy = yx\in\mathfrak{X}$

G.2. $(x+y)z = xz+yz$

G.3. $(xy)z = x(yz)$

G.4. $(\alpha x)y = \alpha(xy)$

Amennyiben a szorzat nem minden (x, y) párra van értelmezve, de ahol igen, ott a G.1.—G.4. tulajdonságok érvényesek, akkor $\mathfrak{X}$ *kvázigyűrű.*

Minden kvázigyűrűben a diszjunkt elemek szorzata θ, amiből következik tetszőleges komponens részlineár esetén, hogy a szorzat vetülete egyenlő a vetületek szorzatával: $\mathrm{pr}\,(xy)=\mathrm{pr}\,x\cdot\mathrm{pr}\,y$.

A dolgozat megértéséhez az itt leírtak elegendők. A részletek iránt érdeklődő, elmélyültebb tanulmányozást igénylő olvasót az irodalomra utaljuk ([1], [2], [3], [4], [5]).

IRODALOM

[1] Акилов, Г. П., Кутателадзе С. С., *Упорядоченные векторные пространства* (Наука, Москва, 1979).
[2] Вулих, Б. З., *Введение в теорию полуупорядоченных пространств* (Москва, 1961).
[3] Vulikh, B. Z., *Introduction to the Theory of Partially Ordered Spaces* (Grpningen, 1967).
[4] Канторович, Л. В., Акилов, Г. П., *Функциональный анализ* (Наука, Москва, 1976).
[5] Канторович, Л. В., Вулих, Б. З., Пинскер, А. Г., *Функциональный анализ в полуупорядоченных пространствах* (Москва, 1950).
[6] Пинскер, А. Г., «Задача линейного программирования в пространствах Канторовича» *ИНИОН деп.* № 1478, Москва, 1977.
[7] Пинскер, А. Г., «Транспортная задача в функциональных пространствах» *Сибирский Мат. Жур.* т. XIX №6 (1978).
[8] Пинскер, А. Г. «Линейная оптимизация в упорядоченных пространствах» *ДАН СССР* т. 242 № 5 (1978).
[9] Пинскер, А. Г., Кузмина, В. В., «Транспортная задача в пространстве полиномов» *ИНИОН деп.* № 1477, Москва, 1977.
[10] Riesz, F., "Sur la decomposition des operations fonctionuelles" *Atti Congresso Bologna* **3** (1928).

(Beérkezett 1979. szeptember 26.)

B. NAGY ANDRÁS
SZÁMÍTÓGÉPALKALMAZÁSI KUTATÓ INTÉZET
1536 BUDAPEST, PF. 227.

ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ В ПОЛУУПОРЯДОЧЕННЫХ ПРОДСТРАНСТВАХ

(Результаты А. Г. Пинскера)

Б. Надь Андраш

В работе излагаются новые результаты линейного программирования в абстрактных пространствах, а именно, задачи, в которых часть коэффициентов являются не числами, а элементами полуупорядоченных (K_σ—) пространств, причем во всех рассматриваемых задачах значения целевой функции принадлежат этому абстрактному пространству, что требует новый подход к понятию оптимальности. Тем не менее, как это доказывается в основной теореме работы, для этих задач имеет место теорема существования оптимального решения аналогично теореме существования для обыкновенных (числовых) задач ЛП. Заодно дается теоретический алгоритм решения этих задач при помощи конечного числа вспомогательных числовых задач.

Помимо основных результатов А. Г. Пинскера, в третей части работы дается разработанная числовая иллюстрация, а так же в приложениях — обзор по элементам теории полуупорядоченных пространств.

Alkalmazott Matematikai Lapok **6** (1980) 123—130

REALIZÁLHATÓ LP ALGORITMUS RÉSZBEN RENDEZETT VEKTORTEREKBEN

B. NAGY ANDRÁS

Budapest

Az [1] dolgozatban ismertettük A. G. PINSZKER eredményeit a lineáris programozás bizonyos irányú általánosítása területén. Nevezetesen arról van szó, hogy az LP feladat egyes paraméterei (változók és/vagy konstansok) nem valós számok, hanem egy absztrakt részben rendezett vektortér elemei. Ugyancsak e vektortér elemei szerepelnek célfüggvény értékek gyanánt is. Az [1] legfontosabb eredménye, hogy megadja az ilyen típusú feladatok megoldhatóságának szükséges és elégséges feltételét, nevezetesen azt, hogy az A és C típusú feladatoknak akkor és csak akkor van optimális megoldása, ha a megengedett megoldások halmaza nem üres és (maximizáció esetén) a célfüggvény e halmazon felülről korlátos. B típusú feladat esetén további követelmény, hogy a *megoldó halmaz* (mely a korlátozó feltételek poliéderének csúcspontjaihoz tartozó valamennyi absztrakt elemet tartalmazza) elemei összehasonlíthatók legyenek a részben rendezett vektortér nulla elemével.

Az [1] második részének végén egy tisztán elméleti értékű algoritmust is közöltünk, mely gyakorlati célokra általában nem alkalmas, miután egy valamelyest is nagyobb feladat esetén a megoldó halmaz meghatározásával kapcsolatos műveletek száma elvégezhetetlenül sokra növekszik.

Jelen dolgozatunkban egy olyan, némileg heurasztikus algoritmust ismertetünk, melynek műveletigénye nagyságrendileg megfelel egy hasonló méretű közönséges LP feladat szimplex-módszerű megoldása műveletigényének, mivel lényegében annak lépéseit követi, és eközben a megoldó halmaznak csak azokat az elemeit kezeli, melyek a megoldás menetében ténylegesen fellépnek. A gyakorlati megoldhatóság tehát csak a részben rendezett tér szerkezetétől függ.

1. Az algoritmus megalapozása

Az algoritmus lényegében a szimplex-módszerre épül. Már az [1]-ben ismertetett algoritmusnak is az a lényege, hogy (A és C típusú feladatok esetén) a feladathoz hozzárendelhető egy

$$P = \left\{ X = (x_1, x_2, \ldots, x_n) \middle| \sum_{j=1}^{n} \alpha_{ij} x_j = b_i \ (i = 1, 2, \ldots, m);\ (x_j, b_i \in \mathfrak{X}) \right\}$$

poliéder, amelynek a komponens altereken vett vetületeinek csúcspontjain haladunk a vetület-feladatokhoz rendelt numerikus LP feladatok szimplex algoritmusai által meghatározott sorrendben. Ugyanis [1] elméleti fejtegetéseinek egyik következménye a következő

MEGÁLLAPÍTÁS. Ha az

$$\text{(A)} \qquad \max \left\{ \sum_{j=1}^{n} \gamma_j x_j \middle| \sum_{j=1}^{n} \alpha_{ij} x_j = b_i \ (i = 1, 2, \ldots, m);\ x_j \in \mathfrak{X}^+;\ (b_i \in \mathfrak{X}^+) \right\},$$

illetve a

$$(C) \quad \max\left\{\sum_{j=1}^{n} c_j x_j \,\middle|\, \sum_{j=1}^{n} \alpha_{ij} x_j = b_i \quad (i = 1, 2, \ldots, m);\ x_j \in \mathfrak{X}^+;\ (c_j \in \mathfrak{X},\ b_i \in \mathfrak{X}^+)\right\}$$

feladatoknak létezik optimális megoldása, akkor az $\mathfrak{X}$ részben rendezett vektortér felbontható végesen sok komponens altérre úgy, hogy az optimumhoz vezető út (azaz a megoldás során a poliéder vetületein bejárt csúcspontok összessége) minden komponens altérben benne fekszik a pozitív kónuszban.
Igaz továbbá a következő nyilvánvaló

MEGÁLLAPÍTÁS. Ha valamely pontok az $\mathfrak{X}$ tér valamely komponens altereinek pozitív kónuszaiban fekszenek, akkor e pontok összevonása az illető komponens alterek összevonásának pozitív kónuszában fekszik. (Legyen $\{x_i\}$ tetszőleges halmaz az $\mathfrak{X}$ térben. Akkor e halmaz összevonása alatt az $\bigcup_i x_i = \sup_i x_i^+ - \sup_i x_i^-$ elemet értjük. Nyilván ha a halmaz elemei diszjunktak, akkor az összevonás azonos az elemek összegével. Alterek összevonása alatt valamennyi reprezentatív halmazuk összevonásainak halmazát értjük és így diszjunkt alterek esetén egyszerűen a direkt összegükről van szó.)

Az [1]-ben közölt algoritmus gyenge pontjai az *a)* és *b)* pontok. Ezek végrehajtása ugyanis gyakorlatilag feltételezi apoliéder valamennyi csúcspontjának meghatározását, ami C_n^m nagyságrendű műveletigényt jelent. Ugyanakkor a fenti megállapítások figyelembevételével az algoritmus jelentős mértékben egyszerűsíthető. Nem kell a megoldó halmaz valamennyi elemét meghatározni és így az $\mathfrak{X}$ teret is elegendő általában kevesebb komponens altérre felbontani. Az [1] harmadik részében szereplő példából is látható, hogy a megoldó halmaz 21 eleméből az optimális megoldás csak hetet tartalmaz, és az eredetileg felvett 6 komponens altér végül is hárommá olvadt össze.

Az egyszerűsített algoritmus lényege a következő. A feladatot úgy kezeljük, mintha közönséges LP feladat lenne egészen addig, amíg valamely elem vagy elemek az $\mathfrak{X}$ térből összehasonlíthatatlanná nem válnak a tér nulla elemével. Ekkor az $\mathfrak{X}$ teret addig bontjuk komponens alterekre, amíg mindezen elemek valamennyi vetülete összehasonlítható nem lesz a nullával és a megoldást az egyes komponens alterekben külön-külön folytatjuk további felbontásig vagy az optimum eléréséig. Az [1] eredményei biztosítják, hogy véges számú lépésben eljutunk az optimális megoldásig, vagy a megoldhatatlanság kritériumáig.

Külön figyelemre méltó, hogy az [1] 2. lemmájában definiált pozitív additív funkcionál tényleges alkalmazására tulajdonképpen nincs is szükség, mivel minden esetben csak pozitív $\mathfrak{X}$-beli elemekkel végzünk műveleteket, és amíg ezekkel nem kell osztani (márpedig nem kell), addig ez nem okoz különösebb problémát.

Mint látható, ebben az algoritmusban a megoldó halmaznak csak a számítások során ténylegesen előforduló elemeit vizsgáljuk és az $\mathfrak{X}$ teret is csak annyi altérre bontjuk, amennyire feltétlenül szükséges. Egyébként a hagyományos szimplex-módszer lépéseit alkalmazzuk (lásd [2]).

2. Numerikus példák

Először az [1]-ben szereplő A típusú szállítási feladatot oldjuk meg a fent elemzett algoritmussal. Induló bázismegoldásnak a következőt választjuk:

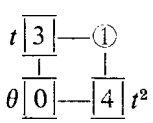
	t^3	t	t^2	t
$2t$	θ_4	$t\,\vert 3$	θ_1	$t\,\vert 0$
t^2	$\theta\,\vert 1$	$\theta\,\vert 0$	$t^2\,\vert 4$	θ_2
t^3	$t^3\,\vert 0$	θ_3	θ_2	θ_5

[0, 3]

Ez a megoldás pozitív a teljes [0, 3] intervallumban.
Kiszámítjuk a potenciálokat:

	0	-1	3	-4
4	0	$\vert 0$	-6	$\vert 0$
1	$\vert 0$	$\vert 0$	$\vert 0$	5
0	$\vert 0$	2	-1	9

Az (1, 3) cellában van a minimális potenciál, tehát itt javítjuk a programot:



Mint látjuk, a $t-t^2$ előjelétől függ az új program megválasztása. Mivel a [0, 1) intervallumban $t-t^2 \geqq 0$, ezért ebben a komponens altérben a t^2 elemet mozgatjuk:

	t^3	t	t^2	t
$2t$	θ_4	$t-t^2\,\vert 3$	$t^2\,\vert 1$	$t\,\vert 0$
t^2	$\theta\,\vert 1$	$t^2\,\vert 0$	θ_4	θ_2
t^3	$t^3\,\vert 0$	θ_3	θ_2	θ_5

[0, 1)

A potenciálok:

	0	−1	−3	−4
4	0	$\overline{\lvert 0}$	$\overline{\lvert 0}$	$\overline{\lvert 0}$
1	$\overline{\lvert 0}$	$\overline{\lvert 0}$	6	5
0	$\overline{\lvert 0}$	4	5	9

itt mindenütt nem negatívak, ezért ebben az altérben elértük az optimumot.

Az [1, 3] intervallumban $t^2-t \geqq 0$, tehát

	t^3	t	t^2	t
$2t$	θ 4	θ 3	t $\overline{\lvert 1}$	$t\overline{\lvert 0}$
t^2	$\theta\overline{\lvert 1}$	$t\overline{\lvert 0}$	$t^2-t\overline{\lvert 4}$	$\theta\overline{\lvert 2}$
t^3	$t^3\overline{\lvert 0}$	θ 3	θ 2	θ 5

[1, 3]

A potenciálok:

	0	−1	3	2
−2	6	6	$\overline{\lvert 0}$	$\overline{\lvert 0}$
1	$\overline{\lvert 0}$	$\overline{\lvert 0}$	$\overline{\lvert 0}$	−1
0	$\overline{\lvert 0}$	4	−1	3

A (2, 4) cella tartalmazza a minimális potenciált (akárcsak a (3, 3)-degeneráció!), tehát itt javítjuk a programot:

$$t\overline{\lvert 1}\text{—}\overline{\lvert 0}\,t$$
$$\quad\;\vdots\qquad\;\vdots$$
$$t^2-t\overline{\lvert 4}\text{—}②$$

A további út t^2-2t előjelétől függ. Mivel az [1, 2) intervallumban $2t-t^2 \geqq 0$, ezért ebben az altérben a t^2-t elemet mozgatjuk:

	t^3	t	t^2	t
$2t$	θ 4	θ 3	$t^2\overline{\lvert 1}$	$2t^2-t\overline{\lvert 0}$
t^2	$\theta\overline{\lvert 1}$	$t\overline{\lvert 0}$	θ 4	$t^2-t\overline{\lvert 2}$
t^3	$t^3\overline{\lvert 0}$	θ 3	θ 2	θ 5

[1, 2)

A potenciálok:

	0	−1	2	1
−1	5	5	$\overline{\lvert 0}$	$\overline{\lvert 0}$
1	$\overline{\lvert 0}$	$\overline{\lvert 0}$	1	$\overline{\lvert 0}$
0	$\overline{\lvert 0}$	4	0	4

mindenütt nem negatívak, tehát itt is eljutottunk az optimumhoz.

Hátra van még a [2, 3] intervallum, ahol $t^2-2t \geqq 0$ és így

	t^3	t	t^2	t
$2t$	θ_4	θ_3	$2t\ \overline{\lvert 1}$	θ_0
t^2	$\theta\ \overline{\lvert 1}$	$t\ \overline{\lvert 0}$	$t^2-2t\ \overline{\lvert 4}$	$t\ \overline{\lvert 2}$
t^3	$t^3\ \overline{\lvert 0}$	θ_3	θ_2	θ_5

[2, 3]

A potenciálok:

	0	−1	3	1
−2	6	6	$\overline{\lvert 0}$	1
1	$\overline{\lvert 0}$	$\overline{\lvert 0}$	$\overline{\lvert 0}$	$\overline{\lvert 0}$
0	$\overline{\lvert 0}$	4	−1	4

A minimális potenciál a (3, 3) cellában van, így itt javítunk:

$$\theta\ \overline{\lvert 1}\text{—}\overline{\lvert 4}\ t^2-2t$$
$$\quad\ \lvert \qquad\ \lvert$$
$$t^3\ \overline{\lvert 0}\text{—}\textcircled{2}$$

Az adott intervallumban $t^3-t^2+2t \geqq 0$, tehát t^2-2t fog elmozdulni:

	t^3	t	t^2	t
$2t$	θ_4	θ_3	$2t\ \overline{\lvert 1}$	θ_0
t^2	$t^2-2t\ \overline{\lvert 1}$	$t\ \overline{\lvert 0}$	θ_4	$t\ \overline{\lvert 2}$
t^6	$t^3-t^2+2t\ \overline{\lvert 0}$	θ_3	$t^2-2t\ \overline{\lvert 2}$	θ_5

[2, 3]

Mivel a potenciálok:

	0	−1	2	1
−1	5	5	[0]	0
1	[0]	[0]	1	[0]
0	[0]	4	[0]	4

valamennyien nem negatívak, ezért itt is megérkeztünk az optimális pontba.

Könnyű ellenőrizni, hogy a kapott teljes megoldás az [1]-ben közöltnek alternatív változata, s a célfüggvény értéke itt is:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} \gamma_{ij} x_{ij}^{0} = \begin{cases} 3t-2t^2, & t\in[0,1) \\ 3t^2-2t, & t\in[1,3] \end{cases}$$

Második illusztrációnk legyen egy standard alakban megadott általánosított A típusú LP feladat, ahol $\mathfrak{X}=R^5$:

$$f(X) = x_1+x_2 \to \max$$

$$x_1-x_2 \leqq (5,3,2,4,5)$$

$$-x_1+x_2 \leqq (3,3,2,1,6)$$

$$2x_1+x_2 \leqq (8,8,5,5,10)$$

$$x_1,\ x_2 \geqq \theta = (0,0,0,0,0)$$

Felírjuk a standard szimplex-táblázatot:

	b					x_1	x_2	u_1	u_2	u_3
$\leftarrow u_1$	5	3	2	4	5	[1]	−1	1		
u_2	3	3	2	1	6	−1	1		1	
$\leftarrow u_3$	8	8	5	5	10	[2]	1			1
f	0	0	0	0	0	−1	−1			

Mivel itt az „első oszlop" valamennyi eleme nem negatív, ezért a szimplex transzformáció szabadon elvégezhető, de a generáló elem megválasztása az egyes komponensekben eltérő szűk keresztmetszet miatt nem egyértelmű. Emiatt az új táblázat „első oszlopa" már nem lesz minden komponensében nem negatív.

	b					x_1	x_2	u_1	u_2	u_3
$\rightarrow x_1$	5	3	2	4	5	1	-1	1		
u_2	8	6	4	5	11		0	1	1	
$\leftarrow u_1$	-2	2	1	-3	0		$\boxed{3}$	-2		1
f	5	3	2	4	5		-2	1		
	*			*						

A kiválasztott új bázismegoldás a csillagokkal jelölt komponensekben nem megengedett, ezért ott majd másik bázismegoldást kell keresni. Egyelőre azonban folytatjuk a számításokat a „jó" komponens altérben.

	b					x_1	x_2	u_1	u_2	u_3
x_1		11/3	7/3		15/3	1		1/3		1/3
$\leftarrow u_2$		6	4		11			$\boxed{1}$	1	0
$\rightarrow x_2$		2/3	1/3		0		1	$-2/3$		1/3
f		13/3	8/3		15/3			$-1/3$		2/3

A vizsgált komponensben az „első oszlop" nem negatív maradt, ezért folytatható a számítás.

	b					x_1	x_2	u_1	u_2	u_3
x_1		5/3	3/3		4/3	1			$-1/3$	1/3
$\rightarrow u_2$		6	4		11			1	1	0
x_2		14/3	9/3		22/3		1		2/3	1/3
f		19/3	12/3		26/3				1/3	2/3

Ez már optimális táblázat, tehát az adott komponens altérben a megoldás:

$$\mathrm{pr}_{\mathfrak{X}_1} x_1^0 = (0,\ 5/3,\ 1,\ 0,\ 4/3)$$
$$\mathrm{pr}_{\mathfrak{X}_1} x_2^0 = (0,\ 14/3,\ 3,\ 0,\ 22/3)$$
$$\mathrm{pr}_{\mathfrak{X}_1} f(X^0) = (0,\ 19/3,\ 4,\ 0,\ 26/3)$$

Most visszatérünk a másik komponens altérhez. Az első táblázatban ezúttal nem az első, hanem a harmadik sorból választunk generáló elemet és azzal végezzük el a transzformációt:

	b					x_1	x_2	u_1	u_2	u_3
u_1	1			3/2			$-3/2$	1		$-1/2$
u_2	7			7/2			$\boxed{3/2}$		1	1/2
$\rightarrow x_1$	4			5/2		1	1/2			1/2
f	4			5/2			$-1/2$			1/2
u_1	16/2			10/2				1	1	0
x_2	14/3			7/3			1		2/3	1/3
x_1	10/6			9/6		1			$-1/3$	0
f	28/6			23/6					2/6	2/6

Itt is eljutottunk az optimális táblázathoz és így ebben az altérben az optimális megoldás:

$$\mathrm{pr}_{\mathfrak{X}_2} x_1^0 = (\ 5/3, 0, 0, \ 3/2, 0)$$
$$\mathrm{pr}_{\mathfrak{X}_2} x_2^0 = (14/3, 0, 0, \ 7/3, 0)$$
$$\mathrm{pr}_{\mathfrak{X}_2} f(X^0) = (19/3, 0, 0, 23/6, 0)$$

A két vetületmegoldás összevonása adja a teljes megoldást:

$$x_1^0 = (\ 5/3, \ 5/3, 1, \ 3/2, \ 4/3)$$
$$x_2^0 = (14/3, 14/3, 3, \ 7/3, 22/3)$$
$$f(X^0) = (19/3, 19/3, 4, 23/6, 26/3)$$

IRODALOM

[1] B. NAGY, A., „Lineáris programozás részben rendezett vektorterekben (A. G. Pinszker eredményei)", *Alkalmazott Matematikai Lapok* 6 (1980).
[2] KREKÓ, B., *Lineáris programozás* (Közgazdasági és Jogi Könyvkiadó, Budapest, 1966).

(Beérkezett: 1980. március 13.)

B. NAGY ANDRÁS
SZÁMÍTÓGÉPALKALMAZÁSI KUTATÓ INTÉZET
1536 BUDAPEST, PF. 227.

РЕАЛИЗУЕМЫЙ АЛГОРИТМ ДЛЯ ЛП В ПОЛУУПОРЯДОЧЕННЫХ ВЕКТОРНЫХ ПРОСТРАНСТВАХ

Б. Надь Андраш

Настоящая работа содержит описание реализуемого алгоритма для задачи обобщенного линейного програмирования в полуупорядоченных векторных пространствах, впервые изученного А. Г. Пинскером. Данный алгоритм является практическим вариантом теоретического алгоритма, вытекающего из доказательства теоремы существования, данного А. Г. Пинскером и воспроизведенного в [1]. Этот алгоритм опирается на обыкновенный симплекс алгоритм и требует произвести немного больше элементарных действий, чем симплекс алгоритм решения обыкновенной задачи ЛП подобного размера.

Alkalmazott Matematikai Lapok 6 (1980) 131 – 143

FÜGGŐSÉGEK RELÁCIÓS ADATBÁZIS MODELLBEN

CZÉDLI GÁBOR

Szeged

Az E. F. CODD [2, 3] által bevezetett funkcionális függőség absztrakt jellemzése W. W. ARMSTRONGtól [1] származik. A dolgozatban újabb függőségi fogalmakat vezetünk be, melyek közül kettőt absztrakt módon is jellemzünk.

1. Bevezetés

Az E. F. CODD [2, 3] által bevezetett relációs adatbázis modell az adatok kezelésének egyik legigéretesebb eszköze. Ez a modell nem a gépi hatékonyságot helyezi előtérbe, hanem az adatokat a felhasználó számára könnyen áttekinthető módon ábrázolja. Az adatok tárolása szemléletes formában, mátrix alakban valósul meg.

A pontos definíció kedvéért legyen Ω egy véges halmaz. Most és a továbbiak során is feltesszük, hogy Ω nem üres. Ω elemeit attributumoknak nevezzük. Minden $a\in\Omega$-ra legyen T_a egy tetszőleges nem üres halmaz. T_a az a attributum értékkészlete, elemeit az a attributum értékeinek nevezzük. Ekkor a $\prod_{b\in\Omega} T_b$ *Descartes-szorzat* tetszőleges véges R részhalmazát Ω feletti relációnak nevezzük. Egy Ω feletti R reláció tehát olyan $g\colon \Omega\to\bigcup\{T_b\colon b\in\Omega\}$ kiválasztási függvényekből áll, ahol $g(a)\in T_a$ $(a\in\Omega)$. Ennek megfelelően egy $\Omega=\{a_1, \ldots, a_n\}$ feletti R reláció az alábbi kétdimenziós táblázattal ábrázolható:

	a_1	a_2	...	a_n
g	$g(a_1)$	$g(a_2)$	...	$g(a_n)$

A hagyományos adatkezelő rendszerekkel analógiát keresve egy $g\in R$ függvényt rekordként képzelhetünk el, melyben $g(a)$ az a attributum értéke a kérdéses rekordban.

A relációs adatbázis modell számára az adatbázis nem más, mint véges sok reláció időben változó rendszere.

2. Függőségek relációkban

A funkcionális függőség fogalmát CODD [2, 3] vezette be az alábbi módon. Legyen $A, B \subseteq \Omega$ és R egy Ω feletti reláció. Akkor mondjuk, hogy B *funkcionálisan függ* A-tól (jelölése $A \xrightarrow[R]{f} B$ vagy röviden csak $A \xrightarrow{f} B$), ha bármely $g, h \in R$ esetén

$$(\forall a \in A)(g(a) = h(a)) \Rightarrow (\forall b \in B)(g(b) = h(b))$$

teljesül.

Minden adattároló rendszernek az a végső feladata, hogy a felhasználó számára aktuális információkat szolgáltasson; a rendszerrel kapcsolatos egyéb tevékenységeket (javítás, törlés, beírás, ...) is ennek érdekében végezzük. Relációs modell esetén az információszolgáltatás kérdése többnyire úgy merül fel, hogy bizonyos adatokat ($A \subseteq \Omega$-ra a $g(a)$, $a \in A$ értékeket) többé-kevésbé ismerve további adatokat ($B \subseteq \Omega$-ra a $g(b)$, $b \in B$ értékeket) szeretnénk megtudni. Az információszolgáltatás lehetősége szoros kapcsolatban áll azzal, hogy milyen mértékben függnek össze az ismert és keresett adatok, azaz milyen függőség van A és B között. Ha például B funkcionálisan függ A-tól, akkor az összes $g(a)$ ($a \in A$) attributum értéket ismerve, biztosak lehetünk abban, hogy a $g(b)$ ($b \in B$) értékeket is megtudhatjuk a tekintett relációból.

Az időben változó relációkkal együtt a függőségek is változnak. Célszerű azonban az adatbázist úgy szervezni, hogy bizonyos információszolgáltatási feladatok mindig könnyen keresztülvihetők legyenek; azaz bizonyos függőségek mindig jelen legyenek. Természetesen minden relációban felléphetnek véletlenszerű, esetleges függőségek is. Az állandó törvényszerű függőségek feltárása nélkül viszont nem mondható, hogy tárolt adatainkat teljesen megértettük.

Dolgozatunkban az adattárolást csak az információszolgáltatás szempontjából vizsgáljuk, ezért a fellépő relációkat időben állandónak tekintjük. Megemlítjük azonban, hogy a relációs adatbázismodell elmélete és alkalmazási területe ennél jóval gazdagabb, és a funkcionális függőségeknek is vannak fontosabb alkalmazásai. Mindezek ismertetése alól DEMETROVICS JÁNOS magyar nyelvű összefoglaló cikke [5] remélhetőleg felment bennünket.

Az alábbiakban három további függőségfogalmat definiálunk. Legyen $A, B \subseteq \Omega$ és R egy Ω feletti reláció. Akkor mondjuk, hogy B *erősen függ* A-tól (jelölése $A \xrightarrow[R]{s} B$ vagy röviden $A \xrightarrow{s} B$) R-ben, ha bármely $g, h \in R$ esetén

$$(\exists a \in A)(g(a) = h(a)) \Rightarrow (\forall b \in B)(g(b) = h(b)).$$

Akkor mondjuk, hogy B *gyengén függ* A-tól R-ben (jelölése $A \xrightarrow[R]{w} B$ vagy $A \xrightarrow{w} B$), ha bármely $g, h \in R$ esetén

$$(\forall a \in A)(g(a) = h(a)) \Rightarrow (\exists b \in B)(g(b) = h(b)).$$

A [4]-ben bevezetett $A \xrightarrow[R]{d} B$ (vagy röviden csak $A \xrightarrow{d} B$) *d-függőség* pedig azt jelenti, hogy bármely $g, h \in R$ esetén

$$(\exists a \in A)(g(a) = h(a)) \Rightarrow (\exists b \in B)(g(b) = h(b)).$$

Az $A \xrightarrow{d} B$ kapcsolatot tehát d-függőségnek nevezzük, és értelemszerűen használjuk az erős függőség (s-függőség), funkcionális függőség (f-függőség) stb. kifejezéseket is. Mielőtt a fent definiált függőségek alkalmazhatóságára próbálunk rámutatni, egy

példát ismertetünk. Legyen $\Omega = \{szerő, cím, terem, polc\}$. Az alábbi táblázatban megadunk egy R Ω feletti relációt, amely egy tizennyolc könyvvel felszerelt könyvtár adatait tartalmazza.

Szerző	*Cím*	*Terem*	*Polc*
1	1	1	2
2	2	1	3
3	3	1	1
4	4	1	2
5	5	2	3
6	6	2	1
7	7	2	2
8	8	2	3
9	9	3	1
10	10	3	2
11	11	3	3
12	12	3	1
1	4	1	1
5	8	3	3
4	1	1	3
7	10	3	2
6	10	2	2
6	9	2	1

A könyvtár három teremből áll, mindegyik teremben három, egyenként két könyv befogadó képességű polc van elhelyezve. A könyvtár úgy van megszervezve, hogy $\{szerző, cím\} \xrightarrow{d}_{R} \{terem, polc\}$. Továbbá $i = 1, 2, \ldots, 12$-re az $\left[\frac{i+3}{4}\right]$-ik terem $\left(1+3\left\{\frac{i}{3}\right\}\right)$-ik polcán található az a könyv, amelynek a szerzője is és címe is i. (Itt $[x]$ és $\{x\}$ az x valós szám egész-, illetve törtrészét jelöli.) A könyvtárba érkező olvasó, aki a keresett könyv szerzőjét vagy címét ismeri — legyen i az ismert attribútum értéke — megtalálhatja a könyvet, ha végignézi az $\left[\frac{i+3}{4}\right]$-ik termet és az $\left(1+3\left\{\frac{i}{3}\right\}\right)$-ik polcokat a másik két teremben. Nem szükséges az egész könyvtárat végignéznie.

A példa kapcsán most megkíséreljük végiggondolni, hogy az újonnan bevezetett függőségek milyen előnyöket rejthetnek magukban egyes konkrét relációk esetén. Mindenekelőtt megjegyezzük, hogy egy $A \to B$ függőséget függvényekkel is megadhatunk. Ha $A \to B$ egy funkcionális függőség, akkor ezt egy $\varphi: \prod_{a \in A} T_a \to \prod_{b \in B} T_b$ (parciálisan értelmezett) függvénnyel adhatjuk meg. Az $A \xrightarrow{f} B$-nek megfelelő információszolgáltatás pedig a φ függvény egy behelyettesítési értékének megadását jelenti. Az információszolgáltatás természetesen a φ fügvény ismerete nélkül is lehetséges az R relációt megtestesítő táblázat végignézésével. Az $A \xrightarrow{d} B$ d-függőséget pedig $\delta_a: T_a \to \prod_{b \in B} T_b$ függvényekkel $(a \in A)$ adhatjuk meg. A δ_a függvényektől azt kell megkövetelnünk, hogy valamely $b \in B$ esetén a δ_a értékének b-ik komponense helyes adatot szolgáltasson. (Gondoljunk a példa kapcsán fellépő $\delta_{szerző}(x) = \delta_{cím}(x) = \left(\left[\frac{x+3}{4}\right], 1+3\left\{\frac{x}{3}\right\}\right)$ függvényekre.)

Mármost a többféle függőségfogalom tanulmányozását az alábbiak indokolják.

(1) A gyakorlat olyankor is felvetheti az információszolgáltatás kérdését, amikor nem ismerjük az összes A-beli attributum értékét, csak legalább egyet. Mint pl. a könyvtár látogatója. Az erős és d-függőség éppen ezzel a szituációval kapcsolatos.

(2) Egy függőséget függvényekkel megadva az információszolgáltatás meggyorsítható. Nem biztos, hogy a keresgélés elkerülhető, de legalább a táblázatnak csak egy részében kell keresgélni.

(3) Ha A és B között különböző függőségek is fellépnek, akkor mód nyílik a legkedvezőbb függőség kiválasztására és függvényekkel történő megadására. Ezek a függvények esetleg — mint példánkban is — egyszerűen megadhatók. Első látásra a funkcionális függőség tűnik a legelőnyösebbnek, hiszen azt függvényekkel megadva elkerülhető a táblázatban történő keresgélés időigényes feladata. Azonban egy „bonyolult" függvényt szintén csak táblázattal tudunk megadni, és így a behelyettesítési érték meghatározása egy másik — esetenként elég terjedelmes — táblázatban történő keresgélést tesz szükségessé. Korábbi példánkban

$$\{szerző,\ cím\} \xrightarrow[R]{f} \{terem,\ polc\}$$

is teljesül, de az ezen funkcionális függőséget megadó függvény értéktáblázata megegyezik R-rel. Gyorsabban megtalálunk egy könyvet a péda ismertetése során leírt módon. Azaz e konkrét esetben célszerűbb a

$$\{szerző,\ cím\} \xrightarrow[R]{d} \{terem,\ polc\}$$

d-függőséget választani.

(4) Esetenként elegendő lehet az is, ha csak egy B-beli attributum értékét tudjuk meg helyesen (nem tudva, hogy melyiket). Pl. ez a helyzet akkor, amikor C-beli attributumok értékeit keressük, B csak közbülső lépés, és $B \xrightarrow[Q]{d} C$ vagy $B \xrightarrow[Q]{s} C$ teljesül egy másik Q relációban. Ez a közvetett információszolgáltatás is hatékony lehet, továbbá a tárolandó adatok vagy függőségeket leíró függvények mennyisége is csökkenhet.

(5) Előfordulhat, hogy A és B között nincs funkcionális függőség, de egy másik típusú függőség fellép.

3. Különböző típusú függőségek kapcsolata

Ebben a fejezetben azt vizsgáljuk, hogy milyen kapcsolatban állnak egymással az egyazon R Ω feletti relációban fellépő különböző típusú függőségek.

3.1. Állítás. Legyen $A, B \subseteq \Omega$ és R egy Ω feletti reláció. Ekkor

$A \xrightarrow[R]{f} B$ akkor és csak akkor teljesül, ha $(\forall b \in B)(A \xrightarrow[R]{w} \{b\})$;

$A \xrightarrow[R]{d} B$ akkor és csak akkor teljesül, ha $(\forall a \in A)(\{a\} \xrightarrow[R]{w} B)$;

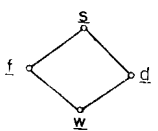
$A \xrightarrow[R]{s} B$ akkor és csak akkor teljesül, ha $(\forall b \in B)(A \xrightarrow[R]{d} \{b\})$;

és

$A \xrightarrow[R]{s} B$ akkor és csak akkor teljesül, ha $(\forall a \in A)(\{a\} \xrightarrow[R]{f} B)$.

Állításunk a definíciók közvetlen következménye, így a bizonyítást mellőzzük.

Vezessük be az $\{f, d, w, s\}$ halmazon a következő $\leqq$ binér relációt[1]: $x \leqq y$ akkor és csak akkor, ha rögzített Ω esetén az R-beli x-függőségek halmaza egyértelműen és R-től függetlenül meghatározza az R-beli y-függőségek halmazát. Nyilván parciális rendezést definiáltunk, melynek diagramja a 3.1. állítás szerint az alábbi:



(Egyszerű ellenpéldával igazolható, hogy f és d a rendezésben összehasonlíthatatlan.)

4. Függőségek absztrakt jellemzése

Tetszőleges R Ω feletti relációra és $z \in \{f, d, w, s\}$-re legyen $\mathscr{Z}_R = \{(A, B) : A, B \subseteq \Omega$ és $A \xrightarrow[R]{z} B\}$. A $\mathscr{Z}_R$ halmazt az R-beli z-függőségek teljes családjának, vagy röviden R z-családjának nevezzük. Minden egyes $\mathscr{Z}_R$ z-család rendelkezik bizonyos tulajdonságokkal. Ezek egy része minden R esetén teljesül (mint pl. $(\forall A \subseteq \Omega)((A, A) \in \mathscr{F}_R)$ a $z = f$ esetben), míg a többi tulajdonság az R reláció specialitása.

Természetesen merül fel az igény, hogy meghatározzuk a z-családok legáltalánosabb — azaz minden R esetén teljesülő — tulajdonságait. Ha egy elméletet szeretnénk kidolgozni relációs adatbázis modellel és z-függőségekkel kapcsolatban, akkor ezt többek között a z-függőségek legáltalánosabb tulajdonságaira — lehetőleg valamennyire — kell alapoznunk. CODD [3] szükségesnek tartotta, hogy számos példát közöljön az általa bevezetett funkcionális függőségen alapuló normálformák magyarázására és definíciói motiválására. A nagyszámú példa részben azt demonstrálta, hogy a normálformák elmélete a funkcionális függőségek legáltalánosabb tulajdonságaira épül. Kevesebb példa is elegendő lett volna, ha a funkcionális függőségek legáltalánosabb tulajdonságai már akkor ismertek lettek volna.

Nem lenne célszerű a z-függőségek valamennyi legáltalánosabb tulajdonságának felsorolására törekednünk. A célnak éppúgy megfelel, ha megadunk néhány olyan tulajdonságot (ún. axiómát) úgy, hogy éppen az axiómák következményei lesznek a z-függőségek legáltalánosabb tulajdonságai. A z-családok absztrakt jellemzésén egy ilyen axiómarendszer megadását fogjuk érteni. Az alábbi három tétel a $z = f$, $z = s$ és $z = d$ esetben nyújt absztrakt jellemzést.

Legyen $P(\Omega)$ az Ω részhalmazainak halmaza. A $P(\Omega) \times P(\Omega)$ *Descartes-szorzat* egy $\mathscr{F}$ részhalmazát absztrakt f-családnak nevezzük, ha $\mathscr{F}$-re az alábbi négy axióma

[1] A reláció elnevezést itt a szokásos értelemben használjuk.

teljesül:

(F1) Minden $A \subseteq \Omega$-ra $(A, A) \in \mathscr{F}$;

(F2) Ha $(A, B) \in \mathscr{F}$ és $(B, C) \in \mathscr{F}$, akkor $(A, C) \in \mathscr{F}$;

(F3) Valahányszor $(A, B) \in \mathscr{F}$ és $A \subseteq C$, $B \supseteq D$, mindannyiszor $(C, D) \in \mathscr{F}$;

(F4) Valahányszor $(A, B) \in \mathscr{F}$ és $(C, D) \in \mathscr{F}$, mindannyiszor $(A \cup C, B \cup D) \in \mathscr{F}$.

4.1. TÉTEL. (ARMSTRONG [1]). Tetszőleges R relációra $\mathscr{F}_R$ egy absztrakt f-család. Fordítva, minden $\mathscr{F}$ absztrakt f-családhoz létezik olyan R reláció, hogy $\mathscr{F} = \mathscr{F}_R$.

A tétel szerint egy funkcionális függőségekre vonatkozó ítélet akkor és csak akkor teljesül minden R relációra $\mathscr{F}_R$-ben, ha következménye az F1, ..., F4 axiómáknak.

A $P(\Omega) \times P(\Omega)$ *Descartes-szorzat* egy $\mathscr{S}$ részhalmazát absztrakt s-családnak nevezzük, ha $\mathscr{S}$-re az alábbi öt axióma teljesül:

(S1) Valahányszor $(A, B) \in \mathscr{S}$, $A_1 \subseteq A$ és $B_1 \subseteq B$, mindannyiszor $(A_1, B_1) \in \mathscr{S}$;

(S2) Ha $(A, B) \in \mathscr{S}$, $(B, C) \in \mathscr{S}$ és $B \neq \emptyset$, akkor $(A, C) \in \mathscr{S}$;

(S3) Valahányszor $(A, B) \in \mathscr{S}$ és $(C, D) \in \mathscr{S}$, mindannyiszor $(A \cap C, B \cup D) \in \mathscr{S}$;

(S4) Valahányszor $(A, B) \in \mathscr{S}$ és $(C, D) \in \mathscr{S}$, mindannyiszor $(A \cup C, B \cap D) \in \mathscr{S}$;

(S5) Minden $a \in \Omega$-ra $(\{a\}, \{a\}) \in \mathscr{S}$.

Hasonlóan, a $P(\Omega) \times P(\Omega)$ egy $\mathscr{D}$ részhalmazát absztrakt d-családnak nevezzük, ha $\mathscr{D}$-re teljesül az alábbi öt axióma:

(D1) Minden $A \subseteq \Omega$-ra $(A, A) \in \mathscr{D}$;

(D2) $\mathscr{D}$ tranzitív, azaz $(A, B) \in \mathscr{D}$ és $(B, C) \in \mathscr{D}$ esetén $(A, C) \in \mathscr{D}$;

(D3) Valahányszor $(A, B) \in \mathscr{D}$, $C \subseteq A$ és $B \subseteq D \subseteq \Omega$, mindannyiszor $(C, D) \in \mathscr{D}$;

(D4) Tetszőleges $(A, B) \in \mathscr{D}$ és $(C, D) \in \mathscr{D}$ esetén $(A \cup C, B \cup D) \in \mathscr{D}$;

(D5) $(A, \emptyset) \in \mathscr{D}$ csak $A = \emptyset$ esetén teljesül.

A továbbiakban az alábbi két tételt fogjuk bizonyítani.

4.2. TÉTEL. Tetszőleges R relációra $\mathscr{S}_R$ egy absztrakt s-család. Fordítva, minden $\mathscr{S}$ absztrakt s-családhoz létezik olyan R reláció, hogy $\mathscr{S} = \mathscr{S}_R$.

4.3. TÉTEL. Tetszőleges nem üres R reláció esetén $\mathscr{D}_R$ egy absztrakt d-család. Fordítva, minden $\mathscr{D}$ absztrakt d-családhoz létezik olyan nem üres R reláció, amelyre $\mathscr{D} = \mathscr{D}_R$.

Megjegyzés. A 4.3. tételből kirekesztettük az $R = \emptyset$ esetet. Ez azonban nem jelenti az általánosság megszorítását, hiszen egyrészt az üres reláció nem túlzottan érdekes, másrészt $\mathscr{D}_\emptyset = P(\Omega) \times P(\Omega)$ miatt $\mathscr{D}_\emptyset$-ra a D1, ..., D5 axiómák közül csak D5 nem teljesül.

5. A 4.2. tétel bizonyítása

A definíciókból közvetlenül adódik, hogy $\mathscr{S}_R$ egy absztrakt s-család. Ezért csak azt kell igazolnunk, hogy minden $\mathscr{S}$ absztrakt s-családhoz található olyan R reláció, hogy $\mathscr{S}=\mathscr{S}_R$.

Legyen $\mathscr{S}$ egy absztrakt s-család, és definiáljunk egy $\psi_{\mathscr{S}}: P(\Omega)\to P(\Omega)$ leképezést a következőképpen:

$$X\psi_{\mathscr{S}} = \{a: a\in\Omega \text{ és } (\{a\}, X)\in\mathscr{S}\} \quad (X\subseteq\Omega).$$

5.1. SEGÉDTÉTEL. A $\psi=\psi_{\mathscr{S}}$ leképezés rendelkezik az alábbi öt tulajdonsággal (tetszőleges $X, Y, Z\subseteq\Omega$-ra):

(P1) $(X\cup Y)\psi = X\psi\cap Y\psi$;

(P2) $X\subseteq Y$ esetén $X\psi\supseteq Y\psi$;

(P3) Ha $Y\neq\varnothing$, $X\subseteq Y\psi$ és $Y\subseteq Z\psi$, akkor $X\subseteq Z\psi$;

(P4) Minden $a\in\Omega$-ra $a\in\{a\}\psi$;

(P5) $\varnothing\psi=\Omega$.

Megjegyezzük, hogy — a segédtételre történő későbbi hivatkozások kedvéért — a felsorolt tulajdonságok nem teljesen függetlenek egymástól, például P5 következik a P2 és P4 tulajdonságokból.

Bizonyítás. Ha $(\{a\}, X)\in\mathscr{S}$ és $(\{a\}, Y)\in\mathscr{S}$, akkor S3 szerint $(\{a\}, X\cup Y)\in\mathscr{S}$. Így $X\psi\cap Y\psi\subseteq(X\cup Y)\psi$. A fordított irányú tartalmazás S1-ből következik, tehát P1 teljesül. P2 szintén S1-ből adódik. Ha P3 premisszái teljesülnek, akkor minden $y\in Y$-ra $(\{y\}, Z)\in\mathscr{S}$. Innen S4 segítségével $(Y, Z)\in\mathscr{S}$ adódik, és S2-ből a P3 következik. S5 miatt P4 triviális. S1 és S5 együttes alkalmazásával $(\{a\}, \varnothing)\in\mathscr{S}$ minden $a\in\Omega$-ra, ahonnan P5 következik.

Egy $\psi: P(\Omega)\to P(\Omega)$ leképezést *s-leképezésnek* nevezünk, ha rendelkezik a P1, ..., P5 tulajdonságok mindegyikével. Egy ilyen ψ s-leképezésre legyen

$$\mathscr{S}_\psi = \{(X, Y): X, Y\subseteq\Omega \text{ és } X\subseteq Y\psi\}.$$

5.2. SEGÉDTÉTEL. Tetszőleges ψ s-leképezésre $\mathscr{S}_\psi$ egy absztrakt s-család.

Bizonyítás. S1 adódik P2-ből, S2 P3-ból, S3 P1-ből, S4 P2-ből és végül S5 P4-ből.

5.3. LEMMA. Rögzített Ω esetén a $\psi\to\mathscr{S}_\psi$ megfeleltetés egy bijektív leképezést létesít az s-leképezések és az absztrakt s-családok halmaza között. Az inverz leképezés nem más, mint az $\mathscr{S}\mapsto\psi_{\mathscr{S}}$ leképezés.

Bizonyítás. Már láttuk, hogy $\mathscr{S}_\psi$ valóban absztrakt s-család és $\psi_{\mathscr{S}}$ valóban s-leképezés. Tetszőleges $X\subseteq\Omega$-ra

$$X\psi_{\mathscr{S}_\psi} = \{a: (\{a\}, X)\in\mathscr{S}_\psi\} = \{a: \{a\}\subseteq X\psi\} = X\psi,$$

azaz $\psi=\psi_{\mathscr{S}_\psi}$. A másik irány bizonyításához vegyük észre, hogy $X\neq\varnothing$ esetén S1 és S4 következtében a $(\forall a\in X)((\{a\}, Y)\in\mathscr{S})$ feltétel ekvivalens az $(X, Y)\in\mathscr{S}$

feltétellel. Ez $X=\varnothing$ esetén is érvényes, hiszen $|Y|\leqq 1$ esetén S1 és S5 következménye, egyébként pedig S3 és S5 folytán $(\varnothing, Y)=(\bigcap_{y\in Y}\{y\}, \bigcup_{y\in Y}\{y\})\in\mathscr{S}$. Ezt felhasználva

$$\mathscr{S}_{\psi_{\mathscr{S}}}=\{(X,Y)\colon X\subseteqq Y\psi_{\mathscr{S}}\}=\{(X,Y)\colon X\subseteqq\{a\colon(\{a\},Y)\in\mathscr{S}\}\}=$$

$$=\{(X,Y)\colon \text{minden } a\in X\text{-re } (\{a\},Y)\in\mathscr{S}\}=\mathscr{S}.$$

Egy R relációhoz definiáljuk a ψ_R s-leképezést: legyen $\psi_R=\psi_{\mathscr{S}_R}$.

5.4. KÖVETKEZMÉNY. A 4.2. tétel bizonyításához elegendő megmutatni, hogy minden ψ s-leképezéshez van olyan R reláció, hogy $\psi=\psi_R$.

Valóban, ekkor tetszőleges $\mathscr{S}$ absztrakt s-családra $\psi_{\mathscr{S}}=\psi_R=\psi_{\mathscr{S}_R}$, ahonnan $\mathscr{S}=\mathscr{S}_R$ következik.

5.5. DEFINÍCIÓ. Legyenek adva az $R_1, \ldots, R_n$ $(n\geqq 1)$ Ω feletti relációk. Ekkor az adott *relációk összegét* az alábbi módon definiáljuk:

$$\sum_{i=1}^{n} R_i=\{(i,g)\colon 1\leqq i\leqq n,\ g\in R_i,\ \text{és valamennyi } a\in\Omega\text{-ra } (i,g)(a)=(i,g(a))\}.$$

Azaz relációk összegét lényegében úgy képezzük, hogy értékkészleteiket indexeléssel diszjunkttá téve vesszük a halmazelméleti uniójukat.

5.6. LEMMA. Legyenek $R_1 \ldots, R_n$ $(n\geqq 1)$ Ω feletti nem üres relációk, A és B pedig Ω részhalmazai. Legyen $R=\sum_{i=1}^{n} R_i$. Ekkor

(a) $A\xrightarrow[R]{s}B$ akkor és csak akkor teljesül, ha $i=1, \ldots, n$-re $A\xrightarrow[R_i]{s}B$ teljesül;

(b) $A\xrightarrow[R]{d}B$ akkor és csak akkor teljesül, ha $i=1, \ldots, n$-re $A\xrightarrow[R_i]{d}B$ teljesül.

A lemma bizonyítását — mely a definíciók közvetlen alkalmazásából áll — elhagyjuk.

5.7. DEFINÍCIÓ. Tetszőleges $A\subseteqq\Omega$-ra egy kételemű R_A Ω feletti relációt definiálunk: $R_A=\{g_A, h_A\}$, ahol $g_A(x)=0$ minden $x\in\Omega$-ra, $x\in A$-ra $h_A(x)=1$ és $x\in\Omega\setminus A$-ra $h_A(x)=0$.

Célunk az, hogy tetszőleges ψ-re az 5.4. következmény szerinti R relációt kételemű relációk összegeként állítsuk elő.

5.8. SEGÉDTÉTEL. Az R_A-hoz tartozó $\psi_A=\psi_{R_A}$ s-leképezés a következő:

$$X\psi_A=\Omega, \quad \text{ha} \quad X\subseteqq\Omega \quad \text{és} \quad X\cap A=\varnothing;$$

$$X\psi_A=A, \quad \text{ha} \quad X\subseteqq\Omega \quad \text{és} \quad X\cap A\neq\varnothing.$$

Bizonyítás: Triviális.

Adott $\psi_i\colon P(\Omega)\to P(\Omega)$ $(i=1, 2, \ldots, n)$ függvényekre definiáljuk a

$\bigwedge_{i=1}^{n}\psi_i\colon P(\Omega)\to P(\Omega)$ függvényt: $X\subseteqq\Omega$-ra legyen $X\bigwedge_{i=1}^{n}\psi_i=\bigcap_{i=1}^{n}X\psi_i$.

5.9. SEGÉDTÉTEL. Legyen R az R_i $(i=1, \ldots, n)$ relációk összege. Ekkor $\psi_R = \bigwedge_{i=1}^{n} \psi_{R_i}$.

Bizonyítás. Az 5.6. lemmát fogjuk elkalmazni. $X \subseteq \Omega$-ra

$$X\psi_R = \{a\colon (\{a\}, X) \in \mathscr{S}_R\} = \{a\colon i = 1, \ldots, n\text{-re } (\{a\}, X) \in \mathscr{S}_{R_i}\} =$$

$$= \bigcap_{i=1}^{n} \{a\colon (\{a\}, X) \in \mathscr{S}_{R_i}\} = \bigcap_{i=1}^{n} X\psi_{R_i} = X \bigwedge_{i=1}^{n} \psi_{R_i}.$$

5.10. SEGÉDTÉTEL. Legyen ψ egy s-leképezés, $A, B \subseteq \Omega$, $A \cap B \neq \emptyset$ és $B\psi = A$. Ekkor az R_A relációhoz tartozó $\psi_A = \psi_{R_A}$ leképezésre $\psi \leqq \psi_A$ (azaz minden $X \subseteq \Omega$-ra $X\psi \subseteq X\psi_A$), de $B\psi = B\psi_A$.

Bizonyítás. Ha $X \cap A = \emptyset$, akkor az 5.8. segédtétel szerint $X\psi \subseteq \Omega = X\psi_A$. Tegyük fel, hogy $c \in X \cap A \neq \emptyset$, és tekintsük az $\mathscr{S}_\psi$ s-családot. Az 5.8. segédtétel szerint $X\psi_A = A$, és így $X\psi \subseteq B\psi$-t kell megmutatnunk. Legyen $x \in X\psi$. Ekkor $(\{x\}, X) \in \mathscr{S}_\psi$, ahonnan S1 szerint $(\{x\}, \{c\}) \in \mathscr{S}_\psi$. Másrészt $c \in A = B\psi$ miatt $(\{c\}, B) \in \mathscr{S}_\psi$. Így S2 szerint $(\{x\}, B) \in \mathscr{S}_\psi$, azaz $x \in B\psi = X\psi_A$. (Az 5.3. lemmát is felhasználtuk.) Tehát $X\psi \subseteq X\psi_A$ teljesül. Végezetül $B\psi_A = A = B\psi$.

5.11. SEGÉDTÉTEL. Legyen ψ egy s-leképezés, és $a \in \Omega$-ra legyen $\psi_a = \psi_{\{a\}\psi}$ az $R_{\{a\}\psi}$ relációhoz tartozó s-leképezés. Ekkor $\psi = \bigwedge_{a \in \Omega} \psi_a$.

Bizonyítás. P4 szerint $\{a\} \cap \{a\}\psi \neq \emptyset$, így $\{a\}\psi = \{a\}\psi_a$ és $\psi \leqq \psi_a$. Ezért a $\psi' = \bigwedge_{a \in \Omega} \psi_a$ jelöléssel élve $\psi \leqq \psi'$ és $\{a\}\psi = \{a\}\psi'$. Ha $X = \emptyset$, akkor $X\psi = X\psi'$ P5 miatt teljesül. Ha $X \neq \emptyset$, akkor P1 szerint (amely s-leképezések metszetére is teljesül)

$$X\psi = \Big(\bigcup_{a \in X} \{a\}\Big)\psi = \bigcap_{a \in X} \{a\}\psi = \bigcap_{a \in X} \{a\}\psi' = \Big(\bigcup_{a \in X} \{a\}\Big)\psi' = X\psi'.$$

Az 5.4. következmény szerint a 4.2. tétel azonnal adódik az 5.9. és 5.11. segédtételekből, hiszen az $R_{\{a\}\psi}$ $(a \in \Omega)$ relációk összegét R-rel jelölve $\psi = \bigwedge_{a \in \Omega} \psi_a = \psi_R$. Ezzel a 4.2. tételt bebizonyítottuk.

6. A 4.3. tétel bizonyítása

A 4.3. tétel és ARMSTRONG tétele (4.1. tétel) között bizonyos dualitás tapasztalható. (Részben a dualitás szó kezdőbetűjéből ered a d-függőség elnevezés, másrészt pedig d a kezdőbetűje az egzisztenciális kvantornak megfelelő diszjunkciónak is.) Ennek köszönhetőleg a 4.3. bizonyításának egy része — körülbelül a 6.7. előtti része — is onnan ered, hogy ARMSTRONG bizonyítását dualizáltuk. A bizonyítás során bizonyos d-családokkal kapcsolatos halmazokra is absztrakt jellemzést adunk.

A definíciókból triviálisan adódik, hogy $\mathscr{D}_R$ egy absztrakt d-család. A fordított irányú állítás igazolásához némi előkészületre lesz szükség.

Vezessük be a $P(\Omega) \times P(\Omega)$ halmazon a következő parciális rendezést:

$(A, B) \leqq (C, D)$ akkor és csak akkor, ha $A \subseteq C$ és $B \supseteq D$. Egy $\mathscr{D}$ absztrakt d-családra jelölje $\mathscr{M}_\mathscr{D}$ a $\mathscr{D}$ maximális elemeinek halmazát. $\mathscr{M}_\mathscr{D}$-t a $\mathscr{D}$ m-családjának fogjuk nevezni.

6.1. SEGÉDTÉTEL. Az $\mathcal{M}=\mathcal{M}_{\mathcal{D}}$ halmaz rendelkezik az alábbi tulajdonságokkal:

(M1) Bármely $A\subseteq\Omega$-ra létezik $(X, Y)\in\mathcal{M}$, hogy $(A, A)\leqq(X, Y)$;

(M2) Ha $(A, B)\in\mathcal{M}$, $(C, D)\in\mathcal{M}$ és $(A, B)\leqq(C, D)$, akkor $(A, B)=(C, D)$;

(M3) Ha $(A, B)\in\mathcal{M}$, $B\subseteq C$ és $(C, D)\in\mathcal{M}$, akkor $A\subseteq C$;

(M4) Ha $(A, \varnothing)\in\mathcal{M}$, akkor $A=\varnothing$.

Bizonyítás. M1, M2 és M4 teljesülése evidens. Tegyük fel, hogy $(A, B)\in\mathcal{M}$, $B\subseteq C$ és $(C, D)\in\mathcal{M}$. Ekkor $(B, C)\in\mathcal{D}$ következik D1-ből és D3-ból, és így D2 miatt $(A, D)\in\mathcal{D}$. D4-et alkalmazva $(A\cup C, D)\in\mathcal{D}$ adódik. Azonban (C, D) $\mathcal{D}$-beli maximalitása folytán $A\cup C=C$, és így $A\subseteq C$.

Nevezzük a $P(\Omega)\times P(\Omega)$ egy $\mathcal{M}$ részhalmazát *m-családnak*, ha rendelkezik az M1, ..., M4 tulajdonságokkal.

6.2. SEGÉDTÉTEL. Legyen $\mathcal{M}$ egy m-család. Ekkor $\mathcal{D}_{\mathcal{M}}=\{(A, B)\colon A\subseteq\Omega, B\subseteq\Omega$ és létezik $(X, Y)\in\mathcal{M}$, hogy $(A, B)\leqq(X, Y)\}$ egy absztrakt d-család.

Bizonyítás. $\mathcal{D}_{\mathcal{M}}$-ben D1, D3 és D5 nyilván teljesül. D2 igazolása érdekében legyen (A, B) és (B, C) $\mathcal{D}_{\mathcal{M}}$-nek két eleme. Ekkor $(A, B)\leqq(A_1, B_1)$ és $(B, C)\leqq$ $\leqq(B_2, C_2)$ teljesül valamely $(A_1, B_1)\in\mathcal{M}$ és $(B_2, C_2)\in\mathcal{M}$ esetén. Minthogy $B_1\subseteq B\subseteq B_2$, M3-ból $A_1\subseteq B_2$ adódik. Ennélfogva $(A, C)\leqq(B_2, C_2)$, és így $(A, C)\in\mathcal{D}_{\mathcal{M}}$.

Ami D4-et illeti, tegyük fel, hogy $(A, B)\in\mathcal{D}_{\mathcal{M}}$ és $(C, D)\in\mathcal{D}_{\mathcal{M}}$. Válasszunk $\mathcal{M}$-ből olyan (A_1, B_1) és (C_1, D_1) párt, hogy $(A, B)\leqq(A_1, B_1)$ és $(C, D)\leqq(C_1, D_1)$. M1 szerint létezik olyan $(U, V)\in\mathcal{M}$, melyre $(B_1\cup D_1, B_1\cup D_1)\leqq(U, V)$. Minthogy $B_1\subseteq U$ és $D_1\subseteq U$, M3 alkalmazható, és azt kapjuk, hogy $A_1\subseteq U$, $C_1\subseteq U$. Így a kívánt $(A\cup C, B\cup D)\in\mathcal{D}_{\mathcal{M}}$ azonnal adódik az alábbiból:

$$(A\cup C, B\cup D)\leqq(A_1\cup C_1, B_1\cup D_1)\leqq(U, V).$$

6.3. LEMMA. Tetszőleges $\mathcal{D}$ absztrakt d-családra $\mathcal{M}_{\mathcal{D}}$ egy m-család. Fordítva, tetszőleges $\mathcal{M}$ m-családhoz pontosan egy olyan $\mathcal{D}$ d-család létezik — mégpedig az előző segédtételben definiált $\mathcal{D}_{\mathcal{M}}$ —, amelyre $\mathcal{M}=\mathcal{M}_{\mathcal{D}}$.

Bizonyítás. D3 szerint $\mathcal{M}_{\mathcal{D}}$ egyértelműen meghatározza $\mathcal{D}$-t. Az állítás többi része az előző két segédtételből adódik.

A most bizonyított lemma szerint absztrakt d-családok helyett m-családokkal is foglalkozhatnánk; elegendő lenne tetszőleges $\mathcal{M}$ m-családhoz egy olyan R relációt találni, amelyre $\mathcal{M}=\mathcal{M}_{\mathcal{D}_R}$. (Az m-családok definíciója azonban még nem elég egyszerű ahhoz, hogy $\mathcal{M}$ függvényében egy ilyen R relációt közvetlenül megadjunk.) A 4.3. tétel igazolása után a 6.3. lemma a maximális d-függőségek családjának absztrakt jellemzését fogja szolgáltatni.

Most bizonyos félhálókat definiálunk. A rövidség kedvéért a $(P(\Omega); \cap)$ 0—1 részfélhálóit (Ω feletti) *d-félhálóknak* fogjuk nevezni. Azaz egy $\mathcal{K}\subseteq P(\Omega)$ halmaz d-félháló, ha $\varnothing, \Omega\in\mathcal{K}$ és $\mathcal{K}$ tartalmazza bármely két elemének halmazelméleti metszetét. A d-félhálók lényeges szerephez jutnak a bizonyításban. Tetszőleges

$\mathcal{M}$ m-családhoz legyen

$$\mathcal{K}_{\mathcal{M}} = \{A\colon A \subseteq \Omega \text{ és } (A, B)\in\mathcal{M} \text{ valamely } B \subseteq \Omega\text{-ra}\}.$$

Egy $\mathcal{D}$ absztrakt d-családra pedig $\mathcal{K}_{\mathcal{D}}$ legyen $\mathcal{K}_{\mathcal{M}_{\mathcal{D}}}$.

6.4. SEGÉDTÉTEL. Tetszőleges $\mathcal{D}$ absztrakt d-családra és $\mathcal{M}$ m-családra $\mathcal{K}_{\mathcal{D}}$ és $\mathcal{K}_{\mathcal{M}}$ d-félhálót alkot.

Bizonyítás. Elegendő azt ellenőrizni, hogy $\mathcal{K}_{\mathcal{M}}$ d-félháló. $\Omega\in\mathcal{K}_{\mathcal{M}}$ adódik M1-ből. M1 és M4 miatt $\emptyset\in\mathcal{K}_{\mathcal{M}}$. Tegyük fel, hogy $A, B\in\mathcal{K}_{\mathcal{M}}$ és válasszunk olyan C, D, U, V halmazokat, amelyekre $(A, C)\in\mathcal{M}$, $(B, D)\in\mathcal{M}$, $(U, V)\in\mathcal{M}$ és $(A\cap B, A\cap B)\leqq(U, V)$. Minthogy $V\subseteq A$ és $V\subseteq B$, M3-at alkalmazva $U\subseteq A$ és $U\subseteq B$ adódik. Így $A\cap B=U\in\mathcal{K}_{\mathcal{M}}$.

6.5. SEGÉDTÉTEL. Tetszőleges $\mathcal{K}$ d-félhálóra

$$\mathcal{D}_{\mathcal{K}} = \{(A, B)\in P(\Omega)\times P(\Omega)\colon (\forall X\in\mathcal{K})(B \subseteq X \Rightarrow A \subseteq X)\}$$

egy absztrakt d-család.

A bizonyítás közvetlen, elhagyjuk.

6.6. LEMMA. Tetszőleges $\mathcal{D}$ absztrakt d-család esetén $\mathcal{K}_{\mathcal{D}}$ egy d-félháló. Fordítva, tetszőleges $\mathcal{K}$ d-félhálóhoz pontosan egy olyan $\mathcal{D}$ absztrakt d-család létezik — mégpedig az előző segédtételben definiált $\mathcal{D}_{\mathcal{K}}$ —, amelyre $\mathcal{K}=\mathcal{K}_{\mathcal{D}}$.

Bizonyítás. Már láttuk, hogy $\mathcal{D}_{\mathcal{K}}$ absztrakt d-család és $\mathcal{K}_{\mathcal{D}}$ d-félháló. Először megmutatjuk, hogy $\mathcal{K}=\mathcal{K}_{\mathcal{D}_{\mathcal{K}}}$. Tegyük fel, hogy $A\in\mathcal{K}$, és válasszunk ki az A részhalmazai közül egy olyan B-t, amely minimális az $(A, B)\in\mathcal{D}_{\mathcal{K}}$ tulajdonságra nézve. $(A, B)\in\mathcal{M}_{\mathcal{D}_{\mathcal{K}}}$ megmutatásához feltesszük, hogy $(A, B)<(C, D)\in\mathcal{D}_{\mathcal{K}}$. Ekkor B választás miatt $A\subset C$, és így $(A, B)<(C, B)\leqq(C, D)$. Így $(C, B)\in\mathcal{D}_{\mathcal{K}}$. Ekkor $B\subseteq A\in\mathcal{K}$ és így $\mathcal{D}_{\mathcal{K}}$ definíciója következtében $C\subseteq A$. Ellentmondást kaptunk, tehát $(A, B)\in\mathcal{M}_{\mathcal{D}_{\mathcal{K}}}$ és $A\in\mathcal{K}_{\mathcal{D}_{\mathcal{K}}}$.

A fordított irányú tartalmazás megmutatásához tegyük fel, hogy $A\in\mathcal{K}_{\mathcal{D}_{\mathcal{K}}}$. Feltehető, hogy $A\neq\Omega$. Ekkor valamely $B\subseteq\Omega$-ra (A, B) maximális $\mathcal{D}_{\mathcal{K}}$-ban. Jelöljük $\mathcal{H}$-val az $\{X\colon A\subset X\subseteq\Omega\}$ halmazt. Minthogy $X\in\mathcal{H}$-ra $(X, B)\notin\mathcal{D}_{\mathcal{K}}$, minden $X\in\mathcal{H}$-hoz létezik olyan $U_X\in\mathcal{K}$, hogy $B\subseteq U_X$, de $X\not\subseteq U_X$. Minthogy $\mathcal{K}$ véges félháló, $H=\bigcap_{X\in\mathcal{H}} U_X\in\mathcal{K}$. Mármost $B\subseteq H$ és $(A, B)\in\mathcal{D}_{\mathcal{K}}$ következtében $A\subseteq H$. Ha H eleme lenne $\mathcal{H}$-nak, akkor $H\not\subseteq U_H$ ellentmondana $H=\bigcap_{X\in\mathcal{H}} U_X\subseteq U_H$-nak. Ezért $A\not\subset H$ és így $A=H\in\mathcal{K}$. Ezzel a $\mathcal{K}=\mathcal{K}_{\mathcal{D}_{\mathcal{K}}}$ egyenlőséget igazoltuk.

Az egyértelműség megmutatásához tegyük fel, hogy $\mathcal{K}=\mathcal{K}_{\mathcal{D}_1}=\mathcal{K}_{\mathcal{D}_2}$. Jelöljük $\mathcal{M}_{\mathcal{D}_i}$-t $\mathcal{M}_i$-vel, $i=1, 2$. Tegyük fel, hoy $(A, B)\in\mathcal{D}_1$, és válasszunk olyan $(A_i, B_i)\in\mathcal{M}_i$ párokat $(i=1, 2)$, hogy $(A, B)\leqq(A_1, B_1)$ és $(B, B)\leqq(A_2, B_2)$. Ekkor $A_2\in\mathcal{K}=\mathcal{K}_{\mathcal{D}_2}$, és így alkalmas C-re $(A_2, C)\in\mathcal{M}_1$. M3-ból $A_1\subseteq A_2$ adódik, ahonnan $(A, B)\leqq(A_2, B_2)$. Így $(A, B)\in\mathcal{D}_2$ következik D3 szerint. Eképpen megmutattuk, hogy $\mathcal{D}_1\subseteq\mathcal{D}_2$, míg a fordított irányú $\mathcal{D}_2\subseteq\mathcal{D}_1$ tartalmazás hasonlóan következik.

Egy $\varphi: P(\Omega) \to P(\Omega)$ leképezést (Ω feletti) *lezárási operátornak* nevezünk, ha tetszőleges $X \subseteq Y \subseteq \Omega$ esetén $X \subseteq X\varphi = (X\varphi)\varphi$ és $X\varphi \subseteq Y\varphi$. Minden egyes $\mathcal{K}$ d-félhálóhoz hozzárendelhetünk egy $\varphi_{\mathcal{K}}$ lezárási operátort a következőképpen: $X \subseteq \Omega$-ra

$$X\varphi_{\mathcal{K}} = \bigcap \{Y: Y \in \mathcal{K} \text{ és } X \subseteq Y\}.$$

Könnyen látható, hogy bármely $X \subseteq \Omega$ esetén $X\varphi_{\mathcal{K}} \in \mathcal{K}$. Továbbá $X \in \mathcal{K}$ akkor és csak akkor teljesül, ha $X = X\varphi_{\mathcal{K}}$.

6.7. LEMMA. Legyen $\mathcal{K}$ egy d-félháló és $X \subseteq \Omega$. Ekkor $X\varphi_{\mathcal{K}} = \{a: (\{a\}, X) \in \mathcal{D}_{\mathcal{K}}\}$.

Bizonyítás. Jelöljük U-val a bizonyítandó egyenlőség jobb oldalát és legyen $\mathcal{D} = \mathcal{D}_{\mathcal{K}}$. D1 és D3 következtében $X \subseteq U$ és $(X, U) \in \mathcal{D}$, míg D4-ből $(U, X) \in \mathcal{D}$ adódik. Legyen A az X-nek egy minimális olyan részhalmaza, melyre $(U, A) \in \mathcal{D}$. Azt állítjuk, hogy $(U, A) \in \mathcal{M}_{\mathcal{D}}$. Tegyük fel ugyanis indirekt, hogy $(U, A) \prec (V, B) \in \mathcal{D}$. Az A választása miatt $U \subset V$ és $(U, A) \prec (V, A) \leqq (V, B)$. Ennélfogva

$(V, A) \in \mathcal{D}$ D3 szerint,
$(A, X) \in \mathcal{D}$ D1 és D3 miatt, és
$(V, X) \in \mathcal{D}$ D2 következtében.

Mármost D3 miatt valamennyi $v \in V$ esetén $(\{v\}, X) \in \mathcal{D}$. Ezért $V \subseteq U$, indirekt feltevésünkkel ellentétben. Azaz $(U, A) \in \mathcal{M}_{\mathcal{D}}$, és így $U \in \mathcal{K}_{\mathcal{D}}$. A 6.6. lemmát alkalmazva kapjuk, hogy $U \in \mathcal{K}$. S minthogy $X \subseteq U$ is teljesül, $X\varphi_{\mathcal{K}} \subseteq U$.

Annak igazolása maradt hátra, hogy $X \subseteq C \in \mathcal{K}$ esetén $U \subseteq C$. Ennek érdekben válasszunk $P(\Omega)$-ból egy olyan D halmazt, melyre (C, D) maximális $\mathcal{D}$-ben. Minhogy (U, A) szintén maximális $\mathcal{D}$-ben és $A \subseteq X \subseteq C$, $U \subseteq C$ azonnal adódik M3-ból.

6.8. KÖVETKEZMÉNY. Legyen $\mathcal{K}$ egy d-félháló és $X \subseteq \Omega$. Ekkor X akkor és csak akkor eleme $\mathcal{K}$-nak, ha $(\{a\}, X) \in \mathcal{D}_{\mathcal{K}}$ csakis $a \in X$ esetén teljesül.

A $\mathcal{K}_1, \ldots, \mathcal{K}_n$ Ω feletti d-félhálók összegén a legszűkebb olyan $\mathcal{K}$ Ω-feletti d-félhálót értjük, amelynek $\mathcal{K}_1, \ldots, \mathcal{K}_n$ részhalmaza. Könnyen igazolható az alábbi

6.9. SEGÉDTÉTEL. Legyen $\mathcal{K} = \sum_{i=1}^{n} \mathcal{K}_i$ a $\mathcal{K}_i$ ($i = 1, \ldots, n$) d-félhálók összege. Ekkor

$$\mathcal{K} = \left\{ \bigcap_{i=1}^{n} A_i : i = 1, \ldots, n\text{-re } A_i \in \mathcal{K}_i \right\}.$$

Valamennyi R relációhoz rendeljük hozzá a $\mathcal{K}_R = \mathcal{K}_{\mathcal{D}_R}$ d-félhálót. Ekkor érvényes az alábbi

6.10. LEMMA. Legyen R az $R_1, \ldots, R_n$ ($n \geqq 1$) nem üres relációk összege. Ekkor $\mathcal{K}_R = \sum_{i=1}^{n} \mathcal{K}_{R_i}$.

Bizonyítás. A $\mathcal{D}_R, \mathcal{D}_{R_i}, \mathcal{K}_R, \mathcal{K}_{R_i}$ helyett rendre a $\mathcal{D}, \mathcal{D}_i, \mathcal{K}, \mathcal{K}_i$ jelöléseket bevezetve tegyük fel, hogy $A \in \mathcal{K}$, és számoljunk az 5.6. és 6.7. lemma, valamint a 6.8. következmény felhasználásával:

$$A = A\varphi_{\mathcal{K}} = \{a: (\{a\}, A) \in \mathcal{D}\} = \{a: i = 1, \ldots, n\text{-re } (\{a\}, A) \in \mathcal{D}_i\} =$$

$$= \bigcap_{i=1}^{n} \{a: (\{a\}, A) \in \mathcal{D}_i\} = \bigcap_{i=1}^{n} A\varphi_{\mathcal{K}_i}.$$

Innen a 6.9. segédtétel segítségével $A \in \sum_{i=1}^{n} \mathscr{K}_i$ adódik. Ezzel beláttuk, hogy $\mathscr{K} \subseteq \sum_{i=1}^{n} \mathscr{K}_i$.

Most azt tegyük fel, hogy valamely i-re $(1 \leqq i \leqq n)$ $A \in \mathscr{K}_i$, de $A \notin \mathscr{K}$. Ekkor van olyan $a \in \Omega$, hogy $a \in A\varphi_{\mathscr{K}} \setminus A$. Ekkor a 6.7. lemma következtében $(\{a\}, A) \in \mathscr{D}$. Az 5.6. lemma alkalmazásával $(\{a\}, A) \in \mathscr{D}_i$ adódik, ahonnan viszont $a \in A\varphi_{\mathscr{K}_i} = A$ következik. Ez pedig $a \in A\varphi_{\mathscr{K}} \setminus A$ miatt lehetetlen. Enélfogva $A \in \mathscr{K}$, és így $\mathscr{K}_i \subseteq \mathscr{K}$. Végül $\mathscr{K}_i \subseteq \mathscr{K}$ $(i = 1, \ldots, n)$ következtében $\sum_{i=1}^{n} \mathscr{K}_i \subseteq \mathscr{K}$.

6.11. SEGÉDTÉTEL. Legyen R_A $(A \subseteq \Omega)$ az 5.7. definícióban megadott kételemű reláció. Ekkor $\mathscr{K}_{R_A} = \{\varnothing, A, \Omega\}$.

A bizonyítást, amely a 6.8. következmény alapján igen egyszerű, az olvasóra hagyjuk.

Az előkészítő lépések után a 4.3. tétel bizonyítása a következő. Legyen $\mathscr{D}$ egy absztrakt d-család, $\mathscr{K} = \mathscr{K}_{\mathscr{D}}$ pedig a $\mathscr{D}$-nek megfelelő d-félháló. Legyen továbbá $R = \sum_{A \in \mathscr{K}} R_A$. Minthogy $\mathscr{K} = \sum_{A \in \mathscr{K}} \{\varnothing, A, \Omega\}$, a 6.10. lemma és a 6.11. segédtétel szerint $\mathscr{K} = \mathscr{K}_R$. Végül a 6.6. lemmát alkalmazva kapjuk a kívánt $\mathscr{D} = \mathscr{D}_R$ egyenlőséget. A 4.3. tételt bebizonyítottuk.

Köszönetnyilvánítás. A szerző őszinte hálával tartozik CSÁKÁNY BÉLA és GÉCSEG FERENC professzoroknak, amiért figyelmét a relációs adatbázis modellre irányították. Szíves tanácsaikért ugyancsak köszönet illeti DEMETROVICS JÁNOST és MAKAY ÁRPÁDOT.

IRODALOM

[1] ARMSTRONG, W. W., "Dependency structures of data base relationships", *Information Processing* 74, North Holland Publ. Comp. 1974, 580—583.
[2] CODD, E. F., "A relational model of data for large shared data banks", *Communications of the ACM*, **13** (1970) 377—387.
[3] CODD, E. F., "Further normalization of the data base relational model", Courant Inst. Comp. Sci. Symp. 6, *Data Base Systems,* Prentice-Hall, Englewood Cliffs, N. J. 1971, 33—64.
[4] CZÉDLI, G., „d-dependency structures in the relational model of data", *Acta Cybernetica* **5** (1980) 49—57.
[5] DEMETROVICS, J., „Relációs adatbázis modell", *MTA SZTAKI Közlemények* 20/1978.

(Beérkezett: 1979. szeptember 30.)

CZÉDLI GÁBOR
JÓZSEF ATTILA TUDOMÁNYEGYETEM BOLYAI INTÉZETE
6720 SZEGED, ARADI VÉRTANÚK TERE 1.

DEPENDENCIES IN THE RELATIONAL MODEL OF DATA

G. CZÉDLI

The relational model of data is one of the most promising tools for handling data. In this model the user's data are represented by relationships. A relationship can be visualized by a matrix whose columns and rows correspond to attributes and records, respectively. It was E. F. CODD who introduced the concept of functional dependency and showed its importance in the relational model of data. This concept was characterized by W. W. ARMSTRONG in an abstract way. In the present paper the author introduces three other concepts of dependencies. Abstract characterizations for two of the new concepts are also given.

Alkalmazott Matematikai Lapok 6 (1980) 145—157

SZIMBOLIKUS VÉGREHAJTÁS ÉS PORGRAMUTAK GENERÁLÁSA

SOÓS KLÁRA

Budapest

A szimbolikus végrehajtásból kiindulva igen sokféle, a programok jobb tesztelését elősegítő eszköz hozható létre. A cikk ismerteti a szimbolikus végrehajtás és a hozzá kapcsolódó szimbolikus végrehajtási fa fogalmát, a létrehozásukkal kapcsolatos problémákat. Ezek után tér ki a szimbolikus végrehajtáson alapuló tesztelési eszközök, módszerek bemutatására. Majd röviden ismerteti az ezeken az elveken alapuló, eddig létrehozott tesztelési eszközöket.

1. Bevezetés

A programkészítés folyamatának lényeges része a program helyességének megmutatása. Napjainkban a program írójának magának kell gyakorlati eszközökkel, módszerekkel igazolnia, hogy programja helyes. Ez általában úgy történik, hogy programját ismerve, megpróbálja olyan bemenő adatokkal „terhelni", futtatni a programját, melyek alapján azt mondhatja, hogy a program lényeges részeit „megmozgatta". Ezt az igazolási folyamatot nevezzük tesztelésnek. A tesztelést különböző automatikus és nem automatikus eszközök segítik ([7], [9], [18], [17]), melyek használata lehetővé teszi, hogy a programozó a lényegesebb problémákra koncentráljon. A továbbiakban a szimbolikus végrehajtás fogalmát és az ezen alapuló tesztelési eszközöket, módszereket ismertetjük.

A következőkben egy egyszerűsített, PL/I-szerű nyelvre definiáljuk, illetve szemléltetjük a szimbolikus végrehajtást és a hozzákapcsolódó fogalmakat. Nem törekszünk matematikai pontosságra. Célunk, hogy a fogalmak lényegét, gyakorlati felhasználását megmutassuk. Ezért a nyelv pontos szintaxisát és szemantikáját nem adjuk meg, a PL/I vagy egyéb ALGOL-szerű nyelvet ismerők számára a nyelvi elemek egyértelműen érthetők. Véleményünk szerint a leírás alapján tetszőleges programozási nyelvre átültethető a szimbolikus végrehajtás, a speciális nyelvi elemek igényelhetnek külön meggondolást.

2. Szimbolikus végrehajtás

A szimbolikus végrehajtás fogalma a normál programvégrehajtás fogalmából természetes módon következik. Bármely programozási nyelv esetén a nyelv pontos szintaxisa és szemantikája alapján lehet a szimbolikus végrehajtás pontos definícióját megadni.

Egy program szimbolikus végrehajtása a program több olyan konkrét végrehajtását tükrözi, melyeknél a végrehajtás menete ugyanaz. Ezekhez a konkrét végrehajtásokhoz tartozó bemenő adatok egy bemenőadat tartományt határoznak meg.

Így erre a bemenőadat tartományra jellemző, hogy bármely elemével mint bemenő adattal a programot végrehajtva a végrehajtás menete ugyanaz lesz. Szimbolikus végrehajtáskor a bemenő változóknak szimbolikus értéket adunk, az értékadásokban ezekre a szimbolikus értékekre hivatkozunk, és végeredményként az eredmény változókra szimbolikus kifejezéseket kapunk értékül. Ha a szimbolikus végrehajtást jól végeztük el, akkor az adott szimbolikus végrehajtáshoz tartozó bemenőadat tartomány bármely elemével végrehajtva a programot, ugyanazt az eredményt kell kapjuk, mintha az eredmény változókra kapott szimbolikus kifejezésekbe helyettesítenénk be a konkrét bemenő értékeket. Ezt a tulajdonságot a szimbolikus végrehajtás kommutativitásának hívjuk.

A szimbolikus végrehajtás menete a következő:

1. kiválasztjuk, hogy a programban milyen út mentén történjen a végrehajtás;
2. definiáljuk, milyen kezdeti szimbolikus értékeket adunk a bemenő változóknak;
3. a kiválasztott utat követve a szimbolikus értékekkel végrehajtjuk a programot.

A végrehajtás lényegében értékadások egymásutánja. (Eljárás- és függvényhívás esetén a bemenő formális paraméterek megkapják az aktuális paraméterek értékét, és ezekkel hajtódik végre a törzs. A formális paraméter tekinthető a bal oldalnak és az aktuális paraméter a jobb oldalnak. Kilépéskor a függvénynév, illetve az aktuális eredmény paraméter tekinthető a bal oldalnak, jobb oldalnak pedig a megfelelő formális paraméter.) Az értékadás jobb oldali kifejezésében szereplő változók helyébe a korábban nyert szimbolikus értéküket helyettesítjük, melyek a bemenő szimbolikus értékek kifejezései is lehetnek. Az így nyert kifejezést esetleg lehet egyszerűsíteni (pl. a konstansokkal való műveleteket el lehet végezni). Az egyszerűsítést csak úgy lehet megtenni, hogy a végrehajtás korábban említett kommutatív tulajdonsága továbbra is érvényben maradjon. A gépi reprezentáció, az egyes részműveleteknél bekövetkezhető kerekítés, alul-, felülcsordulás stb. a lehetséges egyszerűsítések körét lényegesen szűkíthetik, és így az eredmény változók szimbolikus értékei feleslegesen bonyolult, kevésbé áttekinthető kifejezések lehetnek. Ezért a gyakorlatban nem veszik figyelembe ezeket a korlátozó feltételeket. Így a felhasználónak konkrét futtatásokkal kell ellenőriznie, hogy a szimbolikus végrehajtás kommutatív tulajdonsága fennáll-e.

```
1  MAX:PROCEDURE  X,Y ;
2        DECLARE X,Y,Z INTEGER;
3    IF X>Y
4          THEN  Z:=X;
5          ELSE  Z:=Y;
6    RETURN (Z);
7  END;
```

1. ábra
A MAX eljárás

Az 1. ábrán levő egyszerű eljárásnak kétféle szimbolikus végrehajtása lehetséges. Jelölje X szimbolikus értékét α, Y szimbolikus értékét β. Így az eredmény az egyik esetben $Z=\alpha$, a másik esetben $Z=\beta$. A szimbolikus végrehajtáshoz tartozó bemenőadat tartományt az eljárásban szereplő feltétel teljesülése, illetve nem teljesülése határozza meg:

$$\{(\alpha,\beta)|\alpha > \beta\}, \text{ illetve } \{(\alpha,\beta)|\alpha \leqq \beta\}.$$

Ha a programban kiválasztottuk a végrehajtási utat, az eredmény változók szimbolikus értékének kiszámításához az út mentén előforduló elágazási feltételek kiértékelésére nincs szükség. Pontosabban automatikusan azzal a feltételezéssel élünk, hogy a bemenő értékek olyanok, melyekből kiindulva az elágazási feltételek éppen az úton való haladáshoz szükséges igaz vagy hamis értéket veszik fel. Ez azt jelenti, ha egy úton rendre előforduló elágazási feltételekben szereplő változókat (az értékadások végrehajtásához hasonlóan) a korábban nyert szimbolikus értékükkel helyettesítjük, és az így kapott szimbolikus kifejezéseket összegyűjtjük a megfelelő ág követéséhez szükséges logikai (igaz vagy hamis) értékekkel együtt, akkor éppen az út leírását kapjuk. Ha ezeket a szimbolikus kifejezéseket a logikai értékükkel együtt a logikai „és" művelettel összekapcsoljuk, a kapott logikai kifejezés határozza meg, hogy a teljes bemenőadat tartomány mely részhalmazának elemeivel kell futtatni a programot ahhoz, hogy a végrehajtás az adott utat kövesse. Az így nyert logikai kifejezést nevezzük útpredikátumnak. Az útpredikátumok lényeges szerepet játszanak a következőkben ismertetésre kerülő „szimbolikus végrehajtások fájának", valamint a tesztadatok generálásában.

3. Szimbolikus végrehajtások fája

Egy program szimbolikus végrehajtásait az ún. szimbolikus végrehajtások fájával reprezentálhatjuk. A fa minden csúcsa egy végrehajtandó utasítást jelöl, és az utasítások közötti vezérlésátadást egy él jelzi. A csúcsok címkéi az utasítások sorszámai, a gyökérhez a kezdő utasítás tartozik. A csúcsot elhagyó élhez az utasítás végrehajtásának hatására a program állapotában bekövetkező változást rendeljük hozzá. Szimbolikus végrehajtáskor a program állapotát egy adott pillanatban jellemzi:

1. a változók aktuális szimbolikus értéke;
2. az adott pillanatig megtett részút útpredikátuma (ú.p.).

A végrehajtás megkezdésekor ú.p. ≡ igaz, mivel a bemenőadat tartományra semmiféle korlátozást nem teszünk a végrehajtás megkezdése előtt. Egy utasítás hatására az állapotnak általában csak az egyik jellemzője változhat meg. Bizonyos utasítások végrehajtásának (pl. deklarációk feldolgozásának) nincs hatása a program állapotára, ezért ezeket a végrehajtások fájában hely megtakarítása céljából nem tüntetjük fel. Értékadó utasítás hatására a bal oldalon szereplő változó szimbolikus értéke változik meg a szimbolikus végrehajtás leírásánál definiált módon. A feltételes (elágazó) utasítás az útpredikátumot változtathatja meg. A változást a következő formájú IF utasításra definiáljuk:

S: IF ⟨logikai kifejezés⟩
S1: THEN utasítás1;
S2: ELSE utasítás2;

ahol S, S1, S2 értelemszerűen utasítás sorszámok. Ilyen IF utasítás esetén a fát a következőképpen építjük fel:

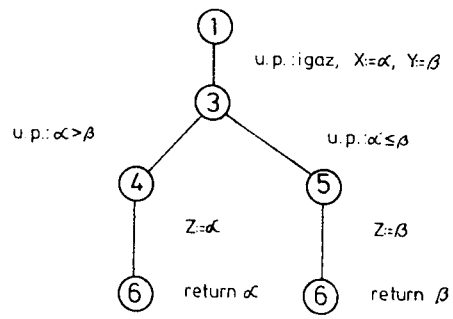
1. Kiszámítjuk a logikai kifejezés szimbolikus értékét, azaz a benne szereplő változókat helyettesítjük a korábban nyert szimbolikus értékükkel. Ha lehet, a nyert kifejezést egyszerűsítjük. A kapott logikai kifejezést B-vel jelöljük.

2. Ha ú.p.$\rightarrow$B$\equiv$igaz, akkor a végrehajtás mindig az utasítás1-gyel fog folytatódni. Ezért az S címkéjű csúcsból egy él indul csak ki, és ez az S1 címkéjű csúcsba vezet.

3. Ha ú.p.$\rightarrow \neg$B $\equiv$ igaz, akkor a végrehajtás folytatódni mindig az utasítás2-vel fog. Így az S címkéjű csúcsból egyetlen él indul ki, és ez az S2 címkéjű csúcsba megy.

4. Ha 2. és 3. egyike sem áll fenn, azaz a végrehajtás az adott útpredikátum esetén a bemenő adatoktól függően mind S1-gyel, mind S2-vel folytatódhat, a végrehajtások fájának is ezt kell tükröznie. S-ből indul egy él az S1 címkéjű csúcsba, melynek címkéje ú.p.$\wedge$B lesz, és indul egy ú.p.$\wedge \neg$B címkéjű él az S2 címkéjű csúcsba. Tehát ebben az esetben a végrehajtások fája elágazik.

A szimbolikus végrehajtások fáját egy egyszerű esetben, a MAX eljárásra mutatjuk be a 2. ábrán.



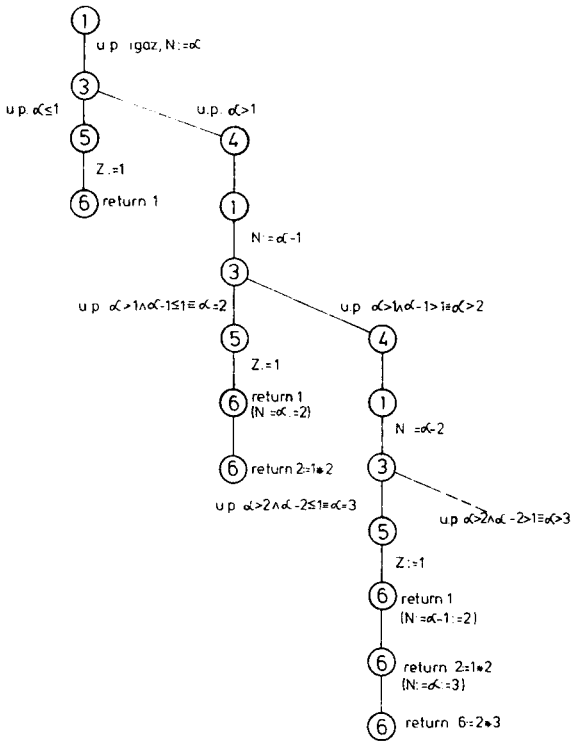
2. ábra
A MAX eljárás szimbolikus végrehajtásainak fája

Az IF utasításhoz hasonlóan definiálhatjuk ciklus utasítás esetén is a fa felépítését. A szimbolikus végrehajtások fája ciklust vagy rekurziót tartalmazó program esetén lehet végtelen. Erre példa a 3. ábra FAKT eljárás végrehajtásainak fája,

```
1  FAKT:PROCEDURE(N);
2        DECLARE  N,Z  INTEGER;
3        IF  N>1
4            THEN  Z:=FAKT(N—1)*N;
5            ELSE  Z:=1;
6        RETURN(Z);
7  END;
```

3. ábra
A FAKT eljárás

melynek egy részletét a 4. ábra mutatja. Ez egyben szemlélteti az eljáráshívás kezelését is. A RETURN utasítás hatására (a normál végrehajtás esetében is) nemcsak visszatérés történik a hívó eljárásba, hanem a nyelv szemantikájának megfelelően bizonyos változók visszanyerik a hívás előtti értékeket. Az áttekinthetőség érdekében a RETURN utasítás ilyen hatását zárójelben tüntetjük fel.



4. ábra
A FAKT eljárás szimbolikus végrehajtásainak részfája

Egy adott program szimbolikus végrehajtásainak fáját a következők jellemzik:

1. A program minden szimbolikus végrehajtásának megfelel egy, a fa gyökerétől valamelyik levélig vezető út, és minden ilyen útnak megfelel egy szimbolikus végrehajtás, azaz a bemenő változóknak tudunk olyan konkrét értéket adni, hogy a végrehajtás az adott utat kövesse.

2. Bármely két különböző levélhez tartozó út különböző végrehajtást reprezentál, hiszen van egy elágazási pont, ahol az útpredikátumok különbözővé válnak.

4. Szimbolikus végrehajtással kapcsolatos problémák

A szimbolikus végrehajtáson alapuló rendszerek egyik célja (l. 6. pont), hogy a program, felhasználó által definiált szimbolikus végrehajtását (végrehajtásait) generálja, és az eredményként kapott útpredikátumot (útpredikátumokat) az eredmény változók hozzá tartozó szimbolikus értékével együtt kiírja a felhasználó részére. Kitérünk néhány, a beolvasó utasításokkal és az adatstruktúrákkal (tömbökkel) kapcsolatos problémára, melyeket a végrehajtásokról kapható információk jobb felhasználhatósága érdekében az eddig létrehozott rendszerek különbözőképpen próbáltak megoldani.

Bemenő változók értéket egy READ utasítás hatására vagy egy eljárás, illetve függvény formális paramétereként kaphatnak. Az utóbbi a normál értékadó utasításhoz hasonlóan kezelhető (l. a szimbolikus végrehajtás leírásánál). READ utasítás esetén a program végrehajtásának megkezdése előtt definiált, a lehetséges szimbolikus elemi értékeket tartalmazó halmazból kell egy elemet az utasításban szereplő változónak értékül adni. Ha ez a READ utasítás többször hajtódhat végre (cikluson belül van), célszerű egyértelműen és szembeötlően jelezni, hogy az új érték ugyanannak a READ utasításnak hatására keletkezett, de ugyanakkor azt is fel kell tüntetni, hogy újabb beolvasás történt. Ezért szokás a szimbolikus értéket „indexelni". Az index utalhat a végrehajtó READ utasítás úton belüli sorszámára (ezt a megoldást alkalmazták a DISSECT szimbolikus végrehajtó rendszernél [12]), vagy arra, hogy a változó a READ utasítás hanyadszori végrehajtásánál kapta az értéket. Az 5. ábrán

```
10  READ FILE(INPUT) INTO(SZAM);
.
.
.
.
70  DO J= 1 TO 5;
71      WRITE FILE(OUTPUT) FROM(SZAM);
72      READ FILE(INPUT, INTO(SZAM);
73  END;
```

5. ábra

```
SZAM    α
        β:1
        β:2
        β:3
        β:4
```

6. ábra

Az 5. ábrán levő programrészlet szimbolikus végrehajtásakor keletkező output

bemutatott programrészlet végrehajtásakor a 6. ábrán látható kiírás jön létre, mely utalhat esetleg egy olyan hibára, hogy az első READ utasításnak is a cikluson belül kellene lennie.

A változó fajták közül az indexelt változókkal foglalkozunk a következőkben. Az egyszerűség kedvéért egydimenziós tömböket tekintünk. Ha az értékadás jobb oldalán levő kifejezésben indexelt változó szerepel és nem határozható meg egyértelműen, hogy a tömb korábban értéket kapott elemei közül melyikre utal az index, akkor az értékadás sem hajtható végre egyértelműen. Tekintsük a V=B(I) alakú értékadást. Ha I értéke nem határozható meg egyértelműen, kétféle megoldás fordul elő az eddigi rendszereknél. Az első módszernél ([2]) az I lehetséges értékeinek megfelelően alternatívákat kell feltételezni, az alternatíváktól függően a szimbolikus végrehajtás több irányban folytatódhat. Az alternatívához tartozó útpredikátumot az alternatívához tartozó feltétellel ki kell egészíteni. A V=B(I) kiértéke-

léséhez feltételezzük, hogy csak a B(i1) és B(i2) elemek kaptak már korábban értéket, és B(i1) szimbolikus értéke b1, B(i2)-é pedig b2. Ekkor a következő esetek lehetségesek:

V új szimbolikus értéke	Az ú.p.-t kiegészítő feltétel
V=b1	I=i1
V=b2	I=i2

A két lehetőségen kívül, ha az útpredikátum valamilyen bemenő érték mellett felveheti az igaz értéket úgy, hogy $I \neq i1$ és $I \neq i2$ is teljesül, akkor nyilván programhiba van. Hisz ekkor létezik olyan bemenő adat, mellyel a programot végrehajtva olyan tömbelemre történne hivatkozás, mely még nem kapott értéket a végrehajtás során. A módszer gyakorlati hátránya, hogy megnöveli a lehetséges szimbolikus végrehajtások számát, és az útpredikátumot is lényegesen bonyolíthatja.

A másik módszer ([12]) nem próbálja meg feloldani a tömb indexből származó többértelműséget. A szimbolikus végrehajtás kiírására utal a program, illetve a végrehajtási út azon pontjára, ahonnan többértelműség származhat. Az ilyen utalásokat a szimbolikus értéktől kettőspont választja el. Az output tartalmazza a változó szimbolikus értékén kívül azon utasításokat is, melyekből származhat a többértelműség. A 7. ábrán levő programrészlet hatására keletkező kiírást a 8. ábra mutatja.

```
20  I:=2;
21  X(2):=0;
22  READ FILE(INPUT) INTO(I);
23  X(I):=10;
24  Y:=X(2);
```

7. ábra

```
:21  X(2):=0
:22  READ FILE(INPUT) INTO(I)
:23  X(I:22):=10
     Y:=X(2):23
```

8. ábra
A 7. ábrán levő programrészlet végrehajtásakor az Y szimbolikus értékének és a kiértékeléshez szükséges utasítások kiírása

5. Szimbolikus végrehajtás és tesztelés

A szimbolikus végrehajtáson alapuló, tesztelést segítő eszközök két nagy csoportba oszthatók. Az egyikbe tartozó eszközök a programot szimbolikusan végrehajtják, és a végrehajtás közben nyert információkat kiírják. A másik csoportba tartozó eszközök a program szimbolikus végrehajtása alapján tesztadatokat generálnak, és ezekkel a programot le is futtatják. A két csoport közötti határ nem éles. Van olyan eszköz, mely mind a két lehetőséget nyújtja a felhasználónak.

Az eszközök alapja a program szimbolikus végrehajtásának generálása. Ezt a feladatot az ún. szimbolikus értelmezők végzik el. Ez történhet statikusan vagy interaktívan. Az interaktív esetben minden elágazásnál a felhasználó mondja meg, hogy a feltétel igaz vagy hamis értéket vegyen-e fel. Tehát ebben az esetben a felhasználó határozza meg, a programnak melyik útja hajtódjon végre szimbolikusan. Ebben az esetben általában nem történik ellenőrzés a végrehajtás alatt, hogy az valódi út-e, azaz létezik-e ténylegesen olyan bemenő érték, melynek hatására a program végrehajtása a kívánt utat követi.

A statikus esetben a program összes, illetve bizonyos feltételeknek eleget tevő szimbolikus végrehajtásai generálódnak. Tetszőleges programozási nyelv esetén nem létezik olyan algoritmus, mellyel tetszőleges program összes szimbolikus végrehajtását generálni lehetne. Egyrészt a korábbiakban már láttuk, a szimbolikus végrehajtási fa végtelen is lehet. Ezért általában a felhasználó írja elő, hogy a bemenő értéktől függő ciklusszámláló esetén a ciklus hányszori végrehajtását vegye figyelembe a szimbolikus értelmező. Másrészt az elágazási pontokban mindig meg kell vizsgálni az útpredikátum és az elágazási feltétel viszonyát (l. az IF utasítás esetén a szimbolikus végrehajtások fájának felépítését). Ha a feltételek szerkezetére nem teszünk kikötéseket, akkor a vizsgálandó formula azonosan igaz volta nem biztos, hogy eldönthető. Ha a feltételben csak két aritmetikai kifejezés valamilyen relációjellel ($<$, $>$, $=$, $\neq$) összekapcsolva szerepelhet, és az aritmetikai kifejezés szimbolikus értéke a bemenő változók lineáris kifejezése lehet csak, akkor a kérdés már eldönthető, mert visszavezethető lineáris egyenlőtlenség rendszerek függetlenségének vizsgálatára. Ezért az eddig létrehozott rendszerek is ilyen korlátozások mellett működnek sikeresen.

A következőkben összefoglaljuk, milyen információkat adnak ki a szimbolikus értelmezők. Egy szimbolikus végrehajtásról megadják, milyen utat követett a végrehajtás, azaz a végrehajtott utasítások sorszámai jelennek meg a kiírásban. Felsorolódnak az elágazásokban előfordult feltételek aktuális szimbolikus kifejezései az út követéséhez szükséges igaz vagy hamis értékkel és az eredmény változók szimbolikus értékei a szükséges kiegészítő információval együtt (l. tömbelem értékadások). A kiírásban olyan hibákról is megjelennek üzenetek, melyek észlelésére a szimbolikus végrehajtás ad lehetőséget (pl. definiálatlan értékű változó). A 10. ábra mutatja a DISSECT által adott kiírást a 9. ábrán levő példa ciklusának kétszeri végrehajtása

```
      DOUBLE PRECISION FUNCTION SIN(X,E)                1
C     THIS COMPUTES SIN(X) TO ACCURACY E                2
      DOUBLE PRECISION E, TERM, SUM                     3
      REAL X                                            4
      TERM=X                                            5
      DO 20 I=3,100,2                                   6
      TERM=TERM*X**2/(I*(I-1))                          7
      IF (TERM.LT.E) GO TO 30                           8
      SUM=SUM+(-1**(I/2))*TERM                          9
   20 CONTINUE                                          10
   30 SIN=SUM                                           11
      RETURN                                            12
      END                                               13
```

9. ábra
SIN eljárás hibával

```
CASE 2: LOOP TWICE
CASE COMMANDS:
8 SELECT F,T;

PATH: 1-10 6-8 11-12
PREDICATES:
  :1  1  DOUBLE PRECISION FUNCTION SIN(X,E)
  :5  8  (X**3/6).GE.E
  :10 8  (X**5/120).LT.E
OUTPUT:
  :1  1   DOUBLE PRECISION FUNCTION SIN(X,E)
  :6  9   NOTE: REFERENCE TO UNINITIALIZED VARIABLE SUM
  :12 12  SIN=?SUM?-(X**3/6)
```

10. ábra
A 9. ábrán levő eljárás szimbolikus végrehajtásakor a DISSECT ezt írja ki

esetén ([12]). A kiírásból egyértelműen kiderül, hogy az eljárás négy hibát tartalmaz. Definiálatlan értékű változóra történő hivatkozást hibaüzenet jelzi. A másik esetben a ciklusból való kilépés feltétele és a SIN változó szimbolikus értékének összehasonlításából kiderül, hogy az utoljára kiszámított, pontosabb értéket eredményező tag nem lett a SIN értékében figyelembe véve. Ez azt jelenti, hogy a függvény 8. és 9. utasításait fel kell cserélni. A további két hiba feltárására a ciklus többszöri szimbolikus végrehajtásakor nyerhető információ. (A 9. sorba $(-1)^{**}$ I/2-t, a 8. sorba (DABS (TERM.LT.E))-t kell írni.)

Ha egy szimbolikus végrehajtáshoz tartozó útpredikátum megoldását vagy megoldásait megkeressük, azaz ha olyan konkrét értékeket keresünk, melyeket a bemenő szimbolikus értékek helyébe írva az útpredikátum igaz lesz, akkor ezeket az értékeket a programnak bemenő adatként adva a végrehajtás az adott utat fogja követni. Ez ad lehetőséget „a program logikájára érzékeny" tesztadatok generálására. Ha az útpredikátum a korábban említett felépítésű (relációk), akkor ez egy egyenlőtlenség rendszer megoldását jelenti. Az egyenlőtlenség rendszer megoldható, ha a relációkban szereplő aritmetikai kifejezések, illetve a megfelelő szimbolikus értékük jól definiált korlátozó feltételeknek (pl. a bemenő változók lineáris kifejezései) tesznek eleget. Ha a program szimbolikus végrehajtásait generáljuk (az említett korlátozásokat figyelembe véve), és minden útpredikátum által meghatározott egyenlőtlenség rendszernek keresünk egy megoldását, az így kapott tesztadatokkal a programot lefuttatjuk, az eredményt ellenőrizzük, akkor azt mondhatjuk, hogy elvégeztük a program minimális tesztelését. A program minden lényeges útját egyszer végrehajtottuk.

A generálandó szimbolikus végrehajtások számának csökkentése, az ellenőrzés könnyítése érdekében a szimbolikus értelmezők a következő lehetőségeket is tartalmazhatják:

1. A felhasználó a bemenő változóknak kezdeti értéket (konkrét vagy relatív szimbolikus értéket) adhat. Ez különösen hasznos lehet, ha a tömb mérete a bemenő adattól függ. Így a kapott szimbolikus értékek is áttekinthetőbbek lehetnek.
2. A felhasználó az útpredikátumnak kezdeti értéket adhat.
3. Korlátozhatja az egy úton előfordulható utasítások számát.

A szimbolikus végrehajtáson alapuló eszközök nem biztosítják a programok megbízható tesztelését. Ennek ellenére két okból is mondhatjuk, hogy az ilyen eszközök segítségével is tesztelt program megbízhatóbb lesz. Egyrészt a felhasználót

szisztematikusabb, fegyelmezettebb tesztelésre kényszeríti. Akár maga jelöli ki a végrehajtási utat, akár azok statikusan generálódnak, a program alapos ismerete szükséges ahhoz, hogy mind a szimbolikus kiírást, mind a tesztadatokkal történt futtatás eredményeit ellenőrizni tudja. Másrészt a szimbolikus kiírás, illetve a tesztadatok automatikus generálásával ezek az eszközök a felhasználó terheit csökkentik, és ezáltal jobban koncentrálhat a problematikusabb esetek vizsgálatára.

Egy programban előforduló hibák három alaptípusát különböztetjük meg ([7]):

1. Az út mentén végzett számítás nem a kívánt eredményt adja. Ezt hívjuk útkiszámítási vagy útfunkció hibának.

2. Az út mentén végzett számítás a kívánt eredményt adja, de az úthoz tartozó bemenőadat tartomány nem egyezik meg a kívánt bemenőadat tartománnyal, mivel valamilyen elágazási feltétel nem jól van megadva. Ezt hívjuk út-tartomány hibának.

3. Az úthoz tartozó tartomány bővebb, mint a kívánt bemenőadat tartomány, mert valamilyen speciális esetre (pl. 0-val való osztás) nem készítettük fel a programot. Ezt hívjuk aleset hibának.

Természetesen a gyakorlatban az alaptípusok kombinációja fordul elő inkább.

Az útfunkció hiba elvileg a szimbolikus eredmények elemzésével feltárható. Gyakorlatban nem biztos, hogy a felhasználó egy, az eredetihez hasonló formában kiírt kifejezésben például a hibás vagy elhagyott zárójelet észreveszi, ha az eredeti formában nem vette észre. Ilyen esetben a generált tesztadattal való futtatáskor kapott eredmény ellenőrzésekor derülhet ki a hiba. Nem ismerhető fel a szimbolikus kiírás elemzésével a gépi alul- és felülcsordulás, kerekítés és a lebegő pontos műveletek elvégzéséből származó problémák. Ezért van feltétlenül szükség a konkrét bemenő adatokkal történő tesztelésre is.

Az út-tartomány hiba feltárását a szimbolikus kiírás elemzése segítheti. A hiba felismerése megköveteli, hogy a felhasználó jól ismerje a programmal szemben támasztott követelményeket. Az ilyen típusú hiba feltárásának a valószínűsége nagyon kicsi a generált tesztadatokkal való futtatás során, mivel az úthoz tartozó tartománynak csak egy eleme kerül kiválasztásra.

Az aleset hiba felismerése talán a legnehezebb. Ha a program írásakor nem gondolt a felhasználó a speciális eset(ek) kezelésére, akkor nem valószínű, hogy a szimbolikus kiírás elemzésekor ez eszébe jut. Az ilyen típusú hibák egy részének felismerésében az értelmező segíthet, ha a megfelelő hiba figyelésére felkészítjük (a tömbindex ellenőrzéséhez hasonlóan). Generált tesztadatokkal történő futtatás során szintén kicsi a hiba feltárásának valószínűsége.

A szimbolikus végrehajtáson alapuló tesztelés megbízhatóságát HOWDEN ([11], [12], [13]) vizsgálta. A különböző tesztelési módszereket hasonlította össze abból a szempontból, hogy a programokban előforduló hibákból az egyes módszerek segítségével hányat lehet megtalálni. Megállapítása szerint egyik eddig ismert tesztelési módszerről sem lehet határozottan megállapítani, hogy az-e a legjobb. Általában a lehető legmegbízhatóbb eredményt több tesztelési módszer párhuzamos alkalmazásával lehet elérni, melyek közé kell tartoznia a szimbolikus végrehajtás eredményeként kapott kiírás elemzésének is, mert ezzel a módszerrel olyan hibák is feltárhatók, melyeket a többi módszerrel kis valószínűséggel lehet megtalálni. Lényegében hasonló következtetésre jut C. GANNON is ([3]).

6. Az eddig létrehozott szimbolikus értelmezők rövid ismertetése

Az EFFIGY interaktív szimbolikus végrehajtó rendszert ([16], [15]) az IBM-nél 1972-ben kezdték létrehozni. PL/I-szerű nyelven írt programok esetén használható. A nyelv a következő elemeket tartalmazza (vázlatosan ismertetjük):

1. külső eljárások a PL/I szerinti paraméter átadási előírásokkal;
2. egész típusú változók, egydimenziós egész típusú tömbök, melyek STATIC vagy AUTOMATIC deklarációban szerepelhetnek;
3. értékadó utasítás, IF utasítás, DO...END közt definiált összetett utasítás, GO-TO utasítás;
4. DO és DO WHILE ciklus utasítások;
5. elemi READ és WRITE utasítás;
6. a következő aritmetikai, logikai műveletek és reláció jelek $+$, $-$, $*$, $/$, $**$, ABS, MOD, $\geqq$, $\leqq$, $<$, $>$, $=$, $\neq$, & (és), | (vagy), $\neg$, $\supset$.

Hibakeresés céljából a következő lehetőségeket tartalmazza:

1. Nyomkövetés. Szimbolikus végrehajtás közben kiírható az utasítás sorszáma, a forráskódú utasítás, a számítási eredmények. Mindezek tetszőleges kombinációja az eljárás vagy a program valamely utasításaira vagy az összes utasítás esetén.
2. Megszakítás. Tetszőleges utasítás végrehajtása előtt vagy után, vagy bármely két utasítás között a szimbolikus végrehajtás megszakítható, és a felhasználó megvizsgálhatja a végrehajtás állapotát, változóknak értéket adhat és folytathatja a végrehajtást.
3. Állapotmentés. A szimbolikus végrehajtás állapotát a felhasználó tárolhatja, és később az adott állapottól folytathatja a végrehajtást.

A bemenő változók értékét a felhasználó írja elő a végrehajtás megkezdésekor. A bemenő érték lehet szimbolikus vagy konkrét érték. Az adott értékkel a végrehajtás interaktívan történik, azaz ha olyan elágazó utasítás kerül végrehajtásra, melynél a feltétel igaz vagy hamis volta az adott bemenő értékkel nem megállapítható, akkor a felhasználó választhat, hogy a feltétel igaz vagy hamis értéket vegyen-e fel, az útpredikátumot újabb feltétellel egészítheti ki. Az EFFIGY nemcsak a teszteléshez ad segítséget, hanem a program helyességének az induktív állítások módszerével történő bizonyításához is ([8]).

A SELECT rendszert ([2]) 1974-ben a *Stanford Research Institute*-nél kezdték kialakítani. Alapja a LISP nyelvnek szűkített változata, mely a következő elemeket tartalmazza:

1. értékadás (SETQ X kif), illetve (SETA tömbnév kif1 kif2) alakban;
2. aritmetikai műveletek
   PLUS, TIMES, DIFFERENCE, QUOTIENT, MINUS, ABS;
3. GO, FOR, WHILE, UNTIL és COND vezérlő utasítások;
4. logikai műveletek és relációk
   AND, OR, NOT, EQ, NEQ, GT, LT, LP, GTQ, LTQ;
5. szubrutin- és függvényhívás.

Az utak generálása automatikusan történik, megállapítja, mely utak nem elérhetők. A szimbolikus végrehajtás után kiírja az eredmény változók szimbolikus értékét. Megpróbál tesztadatot generálni az adott úthoz. Ehhez szükséges, hogy az úthoz tartozó predikátum lineáris egyenlőtlenség rendszert vagy bizonyos típusú nemlineáris egyenlőtlenség rendszert határozzon meg, melyben egész vagy valós

típusú változók fordulhatnak elő. A felhasználó a tesztadat generálást vezérelheti abban az értelemben, hogy az úthoz tartozó tartomány határához közeli vagy ún. centrális adattal kéri a program futtatását. Lehetőség van az EFFIGY-hez hasonlóan helyesség bizonyításra is.

A DISSECT ([12]) rendszer ANSI standard FORTRAN programok tesztelésére készült. A bemenő változók szimbolikus vagy konkrét értékével hajtható végre a program, tesztadatokat nem generál. Az út kiválasztást a végrehajtás megkezdése előtt a felhasználó különböző utasítások segítségével végezheti, egyszerre több utat is generáltathat. A szimbolikus kiírásban a bemenő értékek az úton belüli utasítás sorszámmal „indexelve" jelennek meg, a tömb indexből származó többértelműségre is az utasítás sorszámok utalnak (l. a Szimbolikus végrehajtással kapcsolatos problémák c. pontot).

CLARKE rendszere ([4]) is ANSI FORTRAN programok tesztelését segíti. Az út kiválasztás mind statikusan, mind interaktívan történhet. A szimbolikus végrehajtás során bizonyos hiba ellenőrzésekre is van lehetőség. E célból az útpredikátum ideiglenes feltételekkel egészül ki. A következő módon ellenőrzi a tömb index alsó, illetve felső határon kívül esését. Feltételezi, hogy az index a határokon kívül eshet, és az így kapott egyenlőtlenség rendszert próbálja megoldani. Ha létezik megoldás, akkor előfordulhat ilyen hiba a program végrehajtása során. A rendszer a kiválasztott útra (utakra) generál tesztadatot.

A SMOTL rendszer ([1]) teljes tesztadat halmazt generál SMOD nyelven írt programokhoz. A SMOD COBOL-szerű nyelv, melyben a külső adattárolón levő adatok közvetlen elérése nem lehetséges. Az eddig ismertetett rendszerektől eltérően tehát inkább ügyviteli feladatok megoldására készült programok tesztelésére alkalmas. A rendszer célja automatikusan generálni tesztadatoknak egy olyan halmazát, lehetőleg minimálisat, melynek összes elemével futtatva a programot, minden végrehajtható elágazás legalább egyszer végre is hajtódik.

7. Összefoglalás

A szimbolikus végrehajtás tehát igen hasznos segédeszköze a programtesztelésnek. Nem tértünk ki más területeken való alkalmazási lehetőségeire (pl. program helyesség bizonyítása [8]).

Az eddigi kísérletek is igazolták, a felhasználó számára könnyen kezelhetők, érthetők ezek a rendszerek. Elsősorban logikai hibák feltárására alkalmasak.

Szimbolikus végrehajtást alapul véve sokféle eszköz hozható létre, melyek elősegítik a megbízhatóbb programok készítését. Ezek segítségével a felhasználó a régi tesztelési módszerei (aktuális adatokkal való futtatás) mellett az új lehetőségek alkalmazásával eredményesebben dolgozhat. Olyan rendszerek létrehozását tartjuk célszerűnek, melyek viszonylag kevés felhasználói információ kérésével automatikusan is végeznek útkiválasztást, tesztadat generálását, és ha a felhasználó jól megismerte az így nyert könnyítéseket, maga is átveheti a rendszer vezérlését, a tesztelés irányítását. Minden fordítóprogram mellé hasznos lenne ilyen lehetőségeket adó szimbolikus értelmezőt megvalósítani, még akkor is, ha a korábban említett korlátozó feltételek miatt nem minden felhasználói program szimbolikus végrehajtására és ebből kiindulva tesztadat generálására is lenne alkalmas. Mindenképpen fontos szerepet játszhatnak az ilyen rendszerek a programozás oktatásában, a szisztematikus tesztelésre való

nevelésben. A nagyobb rendszerek tesztelésében részben a korábban említett korlátozó feltételek nehezíthetik, akadályozhatják meg alkalmazását, részben a gépi korlátok. Ez utóbbi csökkentése érdekében tovább kellene fejleszteni az eddigi rendszereket is úgy, hogy a *top-down* program fejlesztést is segítsék.

IRODALOM

[1] BICEVSKIS, J., BORZOVS, J., STRAUJUMS, U. ZARINS, A. and MILLER, E., "SMOTL — A system to construct samples for data processing program debugging", *IEEE Trans. on Soft. Eng.* SE-5 (1979) 60—66.

[2] BOYER, R. S., ELSAPS, B. and LEVITT, K. N., "SELECT — A formal system for testing and debugging programs by symbolic execution", *Proc.* 1975 *Int. Conf. Reliable Software*, 1975, 235—245.

[3] GANNON, C., "Error detection using path testing and static analysis", *Computer* (1979 aug.) 26—31.

[4] CLARKE, L. A., "A system to generate test data and symbolically execute programs", *IEEE Trans, on Soft. Eng.* SE-2 (1976) 215—222.

[5] DARRINGER, J. A. and KING, J. C., „Application of symbolic execution to program testing", *Computer* (1978 apr.) 51—60.

[6] DERVADERICS, K., SÁGHY, A., SOÓS, K. és SZÉPLAKI, Á., „Operációs rendszerek és fordítóprogramok minőségvizsgálata", *SZÁMKI Közlemények*, 18/1978.

[7] GOODENOUGH, J. B. and GERHART, S. L., "Toward a test data selection", *IEEE Trans. on Soft. Eng.* SE-1 (1975) 156—173.

[8] HANTLER, S. L. and KING, J. C., "An introduction to proving the correctness of programs", *Computing Surveys* 8 (1976) 331—353.

[9] HETZEL, W. C., *Program Test Methods* (Englewood Cliffs, New Jersey, Prentice-Hall, 1973).

[10] HOWDEN, W. E., "Methodology for the generation of program test data", *IEEE Trans. on Computers* C-24 (1975) 554—559.

[11] Howden, W. E., "Reliability of the path analysis testing strategy", *IEEE Trans. on Soft. Eng.* SE-2 (1976) 208—215.

[12] HOWDEN, W. E., "Symbolic testing and the DISSECT symbolic evaluation system", *IEEE Trans. on Soft. Eng.* SE-3 266—278.

[13] HOWDEN, W. E., "Theoretical and empirical studies of program testing", *IEEE Trans. on Soft. Eng.* SE-4 293—298.

[14] HUANG, J. C., "An approach to program testing", *Computing Surveys* 7 (1975) 113—128.

[15] KING, J. C., "A new approach to program testing", *Proc.* 1975 *Int. Conf. Reliable Software*, 1975, 228—233.

[16] KING, J. C., "Symbolic execution and program testing", *CACM* 19 (1976) 385—394.

[17] YEH, R. T., *Current Trends in Programming Method Methodology* (Englewood Cliffs, New Jersey, Prentice-Hall, 1977).

[18] VÁRKONYI, Zs., *Bevezetés a modern programtesztelésbe* (Műszaki Könyvkiadó, Budapest, 1979).

(Beérkezett: 1980. február 21.)

SOÓS KLÁRA
SZÁMÍTÓGÉPALKALMAZÁSI KUTATÓ INTÉZET
1536 BUDAPEST, PF. 227.

SYMBOLIC EXECUTION AND THE GENERATION OF PROGRAM PATHS

K. Soós

Several software tools based on symbolic execution can be built for helping the better testing of programs. In this paper a review of the notion of symbolic execution, the symbolic execution tree and the problems of their generation is given. Then the testing tools and methods based on symbolic executions are discussed. At the end a short review of the testing tools having been built on the basis of these principles is presented.

A KÜLFÖLDI SZAKIRODALOMBÓL

PROGRAMOZÁSI NYELVEK — FOGALMAK ÉS KUTATÁSI IRÁNYOK[1]

PETER WEGNER

„Véleményem szerint jelenleg a határán vagyunk annak, hogy végre felfedezzük, milyennek kellene lenniük a programozási nyelveknek. Azt várom, hogy a következő néhány évben számos megbízható kísérletet fogok látni a programozási nyelvek tervezésével kapcsolatban, és arról álmodom, hogy 1984-ig egy jó koncepciójú fejlesztés fog kibontakozni, egy valóban jó programozási nyelvnek, vagy — ami még valószínűbb — a nyelvek egy összefüggő családjának az előállítására. Azt hiszem, hogy az emberek úgy ki fognak ábrándulni azokból a nyelvekből — még a Cobolból és a Fortranból is —, amelyeket most használnak, hogy ez az új nyelv, az Utópia 84, valóra fog válni. Jelenleg még messze vagyunk ettől a céltól, de már vannak jelek, hogy egy ilyen nyelv körvonalai lassan kezdenek kirajzolódni." KNUTH [36, 263. oldal].

„Az összetett jelenségek megértésének a kifejlesztésében az emberi értelem számára rendelkezésre álló leghatékonyabb eszköz az absztrakció. Az absztrakció a valóságos világ bizonyos tárgyai, helyzetei és folyamatai közötti hasonlóságok felismeréséből, továbbá abból a döntésből származik, hogy ezekre a hasonlóságokra koncentráljunk, és a különbségeket egyelőre hanyagoljuk el." HOARE [32, 83. oldal].

A cikk négy fejezetből áll. A bevezető fejezetet „A programozási nyelvek fejlődése", „Nyelvi fogalmak", és „Kutatási irányok" című fejezetek követik. A bevezető fejezet részletesen tárgyalja a programozási nyelvek kutatásának céljait, figyelembe veszi az objektumok és a tevékenységek közötti összefüggéseket, elemzi az absztrakció fogalmát és szerepét a programozási nyelvek kutatásában.

A programozási nyelvek fejlődését a következők szerint írjuk le: *első generációs nyelvek* (Fortran I, Algol 58, Flomatic, IPL 5), amelyeket 1950—58 között fejlesztettek ki; *második generációs nyelvek* (Fortran II., Algol 60, Cobol, Lisp), amelyeket 1959—61 között hoztak létre; *harmadik generációs nyelvek* (PL/1, Algol 68, Snobol 4, Simula 67, Pascal, APL, Basic), amelyeket az 1962-től 1969-ig terjedő időszakban fejlesztettek ki.

Bemutatjuk mindegyik nyelv jellemző tulajdonságainak lényegét azért, hogy a nyelvek tulajdonságaira vonatkozó tervezési kérdéseket egymásután nyelvtől független módon tárgyalhassuk.

A nyelvi fogalmakat tárgyaló fejezet célja azoknak a nyelvi mechanizmusoknak az azonosítása, amelyek lehetővé teszik a programozó számára, hogy a végrehajtandó kiszámítás szempontjából kedvező absztrakt viselkedési formákat fogalmazzon meg. E fejezetben tárgyalt absztrakciós mechanizmusok felölelik a *típusokat* (amelyek az objektumokhoz rendelt absztrakt viselkedést definiálják), a *vezérlési struktúrákat* (amelyek a tevékenységek sorrendi viselkedésének sémáit definiálják) és a *modulokat* (amelyek lehetővé teszik a felhasználóknak, hogy a viselkedés új, összetett formáit definiálja).

A kutatási irányokkal foglalkozó fejezet áttekinti mind a nyelv tervezésének témakörét, mind pedig az elméleti kutatási területeket. A tekintetbe vett nyelvtervezési témakörök felölelik a tárgymodellezés, összekapcsolás *(binding)*, típusdefiníció, beburkolás *(encapsulation)* és vezérlés témaköreit. A figyelembe vett elméleti területek magukba foglalják a specifikáció, verifikáció, programtranszformáció, programszintézis és szemantika problémáit.

[1] Ez a tanulmány "Programming languages — concepts and research directions" címmel a *Research Directions in Programming Methodology* (MIT Press, 1979) könyvben, a 425—488. oldalakon jelent meg. A magyar nyelvű fordításhoz a szerző hozzájárult.

1. Bevezetés

A programozási nyelvek területén folyó kutatások nagy volumenét jól illusztrálja négy, a közelmúltban kizárólag a programozási nyelveknek szentelt konferencia kiadványa [1, 2, 3, 4], az évenkénti POPL konferencia anyaga [5], és más aktuális írásművekben [6, 7, 8] a programozási nyelvekre vonatkozó cikkek nagy száma. A programozási nyelvek területén folyó gyakorlati és elméleti kutatások spektruma rendkívül változatos és a tárgyalásnak több szintjét kell figyelembe venni. Az absztrakció fogalma azonban egységes vezérfonalat nyújt a tárgyaláshoz.

A programozási nyelvek egyidejűleg absztrahálnak a számítógépekből, amelyeken azokat megvalósítják és az alkalmazásokból, amelyeknek a modellezésére tervezték azokat. A számítógépektől nyert absztrakció meghatározza az implementációs modellt (tárgymodellt), amely típustól független absztrakciókat rögzít, olyanokat, mint változók, hozzáférés, értékadás, hatáskör, élettartam, objektum-létrehozás és -törlés, valamint típusdefiníció. Az alkalmazásokból nyert absztrakció a speciális adattípusok típus-függő műveleteit határozza meg.

A programozási nyelvek kutatása felöleli a nyelvek tervezésével kapcsolatos kutatásokat, valamint az elméleti, implementációs és optimalizációs kutatásokat. Hangsúlyozni kívánjuk a nyelvtervezési kutatásokat, és a nyelv tervezése szempontjából lényeges elméleti kutatásokat olyan területeken, mint a specifikáció, verifikáció és szemantika. A fordítóprogramok készítésének tárgykörében folyó nagyarányú kutatásokat, a szintaktikai elemzést és az optimalizálást nem tárgyaljuk.

1.1. *A programozási nyelvekkel kapcsolatos kutatások céljai*

A nyelvtervezési célokat befolyásolja a számítógéprendszerek változó jellege és a számítógépek alkalmazásai.

A számítógéprendszerek fejlődésében a nyelv tervezésére nagy hatást gyakorolt az időosztás, multiprocesszálás, miniszámítógépek, mikroszámítógépek, osztott hálózatok, osztott adatbázisok, real-time korlátozások közé beágyazott számítógépek stb. megjelenése. Ezek a fejlesztések mind nyelvtervezési, mind pedig implementációs kutatásokat egyaránt létrehoztak.

Az elmúlt tíz év során az alkalmazások mind tartalmukban, mind méretükben sokat változtak. A tartalmi változások magukban foglalják az erősen algoritmikus jellegű numerikus alkalmazásoktól az adatkezelést igénylő alkalmazások felé való eltolódást. A méreti változások a programozási nyelveknek a software fejlesztésben betöltött szerepét változtatták meg. A programokat többé nem tekintjük fekete dobozoknak, amelyeket nem kell felnyitni, miután egyszer átadták azokat, hanem folyamatosan változó struktúráknak, amelyeknek nemcsak megbízhatónak és hatékonynak kell lenniük, de jól dokumentáltnak és olvashatónak is, hogy könnyű legyen azokat karbantartani és továbbfejleszteni. A programozási nyelvek eszközök a software termék bonyolultságának a kezelésére, annak életciklusa során és nem csupán jelölésformák az algoritmusok kifejezésére. Az olyan általános nyelvtervezési céloknak, mint a megbízhatóság, karbantarthatóság és bizonyíthatóság, a speciális nyelvi jellemzőkre gyakorolt hatását jól szemlélteti a DOD1 közös nyelvi munkacsoportnak az „Ironman" specifikációja [9] és az Euclid nyelv [10].

A programozási nyelvek kutatásának jelenlegi céljai a következőket ölelik fel:

1. Objektum-definíciós (tudás-reprezentációs) lehetőségeknek a kifejlesztése, hogy ugyanazon az absztrakciós szinten, amelyiken az algoritmust kezeljük, definiálni tudjunk adatobjektumokat és azokkal műveleteket tudjunk végezni.

2. A modularitás mélyebb megértése és nyelvi támogatás nyújtása a modularitás számára, hogy a moduláris komponensekből szisztematikus módon nagy rendszereket lehessen felépíteni.

3. A „megbízható komponensek technológiájának" a kifejlesztése, felhasználva a strukturált programozást, specifikációt, verifikációt, programkipróbálást és a programozási módszertan más eszközeit.

4. Nyelvi struktúrák és nyelvi modellek elméleti tanulmányozása, egyrészt azért, mert ez a megismerés egyik alapvetően érdekes részterülete, amelynek eredményei hatnak a matematikára, a nyelvészetre és a filozófiára, másrészt azért, mert egy ilyen tanulmányozás jelentős ismeretet és eszközt nyújthat gyakorlati célok eléréséhez is.

Az említett első cél a programozási nyelvek kifejező erejének egy fontos irányban történő növelésére vonatkozik. A második és a harmadik cél azzal kapcsolatos, hogy a programozást „nagyban", illetőleg „kicsiben" [11] szilárd alapokra helyezzük. A negyedik cél egy szimbiotikus kapcsolatot alkot az első három céllal abban, hogy az elméleti kutatás adja a szilárd alapot a gyakorlati eszközök és technikák számára, míg a gyakorlati alkalmazás adja az indítékot és biztosítja a konkrét modellek forrásait az elméleti munka számára.

A számítástechnikai alkalmazások a *Parkinson-törvényt* követik abban, hogy azok skálája szétfeszíti a technológia által megszabott korlátokat. Létezik egy olyan gazdasági „törvény", hogy a rosszul strukturált rendszerek irányíthatósági komplexitása *(management complexity)* azok méretével exponenciálisan növekszik és valóban megfigyelhettük, hogy a programozási rendszerek méretének és arányainak megnövekedése a komplexitásnak egy olyan kezelhetetlenségét okozta, amely a programrendszerek előállításának legfontosabb szűk keresztmetszetét képezi ma. Egy „ideális" hierarchikusan strukturált rendszerben a komplexitás bármely döntési ponton a rendszer méretétől független állandó kell, hogy legyen, amelyet csak a döntésnél közvetlenül érintett modulok komplexitása és száma határoz meg, a teljes komplexitás a rendszer n méretének lineáris vagy legfeljebb $n \log n$ nagyságrendű függvénye lehet. Olyan moduláris nyelvek kifejlesztése, amelyek vezérelhetővé teszik a „*management*" komplexitásban az exponenciális növekedést, a programozási nyelvi kutatások egyik fontos célját képezik.

A programozási nyelvek, a természetes nyelvekhez hasonlóan, hatást gyakorolnak a gondolkodási folyamatunkra. Olyan programozási nyelvek, mint a Pascal és az Euclid tervezőiknek azt a felismerését tükrözik, hogy a nyelveknek elő kell segíteniük a matematikai problémamegoldás gondolati folyamatait. A jelenlegi magas szintű nyelvek hatékony problémamegoldó eszközt nyújtanak, de ugyanakkor korlátozzák is a kifejezésmódunkat. Minden nagyobb programozási nyelv megteremtette a saját programjainak a „irodalmát" és az odaadó felhasználóknak egy rezervátumát, akik inkább harcolnak, mint hogy változzanak, mivel ők a gondolkodás és a munka egy létező sablonjának a foglyai. Egy új közös programozási nyelvvel szembeni ellenállásnak az okai hasonlók azokhoz az okokhoz, amelyek miatt az eszperantónak, mint közös természetes nyelvnek a bevezetése ellenállásba ütközik és az ilyen ellenállást nehéz lesz leküzdeni.

1.2. *Tevékenységek és objektumok*

A programozási nyelvet jellemezhetjük objektumaival és primitív tevékenységeivel, továbbá azokkal a mechanizmusokkal, amelyek a primitiv tevékenységeket és objektumokat kiszámítási struktúrákban egyesítik. A programozási nyelvek tevékenységeket összekapcsoló mechanizmusai felölelik a kifejezéseket, az utasításokat, a vezérlési struktúrákat és az eljárásokat. Az objektumokat összekapcsoló mechanizmusok közé a tömbökre, rekordokra és file-okra vonatkozó, adatokat strukturáló mechanizmusok, továbbá az adatabsztrakciók definiálására szolgáló beburkoló mechanizmusok tartoznak.

Különbséget tehetünk tevékenységre orientált, illetve objektumra orientált szemléletű programozás és programozási nyelv között. Az előbbire az Algol 60, az utóbbira a Cobol és az adatbáziskezelő nyelvek [12] szolgáltatnak példát. A tevékenységre orientált szemlélet esetén a programok (algoritmusok) az alapvető mennyiségek és az *objektumok* olyan segédmennyiségek, amelyekkel a program műveleteket végez. Az objektumra orientált szemléletnél viszont az objektumok az alapvető mennyiségek és a *tevékenységek* olyan segédmennyiségek, amelyek az objektumok viselkedésének leírásához szükségesek. A tevékenységre orientált szemlélet a tudományos számítások céljának felel meg, ahol az algoritmusban és az adatszerkezetekben meglevő komplexitás eltűnhet, miután egyszer az eredményt kiszámították. Az objektumra orientált szemlélet az adatkezelésnek, beágyazott számítógépes és más alkalmazásoknak felel meg, ahol az adatszerkezetek a rendszer állapotát képviselik, amely hosszabb ideig létezik és visszahat a rendszer környezetére. A programozási nyelvek fejlődésében az 1950-es és az 1960-as években a tevékenységre való orientáltság dominált. A problémák, a szűk keresztmetszetek és a nagy rendszerek kutatási kérdései az 1970-es években viszont egyre inkább adatszervezési és objektumábrázolási problémákként jelentkeznek.

Az objektumok modellezésével kapcsolatos kérdések a programozási nyelvek kutatásán belül alapvető fontosságúak. A magasabb szintű nyelvekben az objektumok „kanonikus" *(Neumann-elvű)* modellje, amelyre az azonosítók elérési és értékadási tulajdonságai nyújtanak példát, lényegében a memóriarekeszek viselkedéséből nyert absztrakció. A másik objektummodell, az adatfolyam modell, a memóriarekeszek fogalmát olyanokkal helyettesíti, mint a „*pipeline*" vagy objektumfolyam, amelybe az objektumok beléphetnek egy nem destruktív értékadó művelettel, és amelyből azok egy destruktív hozzáférési művelettel kiemelhetők. Egy harmadik modell (az adatfolyam, üzenettovábbító modell) az objektumokat aktív ágenseknek tekinti, amelyek bemenő üzeneteket kaphatnak egy bemenő folyamból, és átalakíthatják azokat kimenő üzenetté egy kimenő folyam számára. Ez a harmadik modell lényegében helyettesíti a passzív memóriarekeszek fogalmát az aktív mikroprocesszorok fogalmával. Ez azért is megfelelő, mivel a mikroprocesszorok előállítása kezd olyan olcsó lenni, mint a memóriarekeszek előállítása volt húsz évvel ezelőtt. Még nem tudjuk, hogyan lehet általános célú számítógépeket felépíteni ilyen komponensekből, és hogyan lehet a számításoknak ilyen modelljén alapuló programozási nyelveket megtervezni. Az objektumra orientált tudás-reprezentációs nyelvek (lásd a 4.2.1. pontot), sokkal természetesebben modellezhetők az adatfolyam architektúrákkal, mint a *Neumann-architektúrákkal* és serkentően hatnak az adatfolyam számítógép architektúrákra. JOHN BACKUS 1978-ban megtartott „*Turing-előadása*" [15] tartalmaz néhány érdekes gondolatot az objektummodellek programozási nyelvekre gyakorolt hatásról.

A programozási nyelvek kutatásának szüksége van egy kifejezetten absztrakt modellre, mind a tevékenységek, mind pedig az objektumok számára. A tevékenységek függvényekkel absztrakt módon modellezhetők, míg az objektumokat algebrák segítségével modellezzük. Parciális rekurzív függvényekkel modellezett tevékenységek intenzív kutatás tárgyát képezik, amióta azt az 1960-as évek elején McCarthy [13] a kiszámíthatóság matematikai elméletének alapjaként javasolta. Az a realizáció, hogy algebrák adják az objektumok matematikai modelljeit, párhuzamosan a függvényekkel, mint a tevékenységek modelljeivel, viszonylag újabb keletű. Az objektum-specifikációs mechanizmus továbbfejlesztése, mind absztrakt szinten, algebrai struktúrák segítségével, mind programozási szinten, olyan moduláris specifikációk segítségével, amelyek különválasztják a felhasználói környezetet az implementációs környezettől, a kutatás fontos területét képezi.

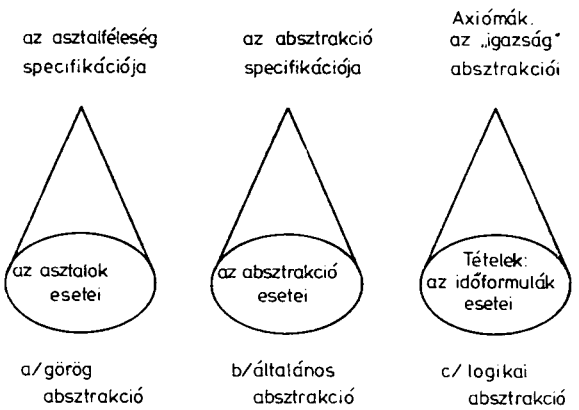
A tevékenységek és az objektumok absztrakt fogalmak, amelyek egy $f(x)$ kifejezés — amely egy f operátornak (függvénynek) egy x argumentumra való alkalmazásából áll —, különböző módon képezett absztrakcióinak felelnek meg. Az $f(x)$-hez rendelt absztrakt f tevékenységet úgy kapjuk meg, ha tekintetbe vesszük az összes, ugyanahhoz az f-hez tartozó kifejezéseket. Ezt jelölhetjük $\lambda x[f(x)]$-szel. Az $f(x)$-hez rendelt absztrakt x objektumot hasonlóképpen úgy kapjuk meg, ha tekintetbe vesszük az összes $f(x)$ formájú kifejezéseket az x-re alkalmazott összes f művelettel. Ezt $\lambda f[f(x)]$-szel jelöljük. Így a tevékenység-absztrakciók az összes olyan x argumentumra kiterjedő kvantálással adódnak, amelyre egy adott f függvény alkalmazható, míg az objektum-absztrakciók egy adott x objektumra alkalmazható összes függvényre kiterjedő kvantálással nyerhetők.

Az individuális $f(x)$ operátor-operandus pároktól a $\lambda x[f(x)]$ absztrakt függvényig vezető absztrakció folyamatát Dijkstra végrehajtási absztrakciónak nevezi [60] és ezt a lambda kalkulus tanulmányozza formális eszközökkel [14], ahol azt funkcionális absztrakciónak nevezik. Az objektum-absztrakciót nem tanulmányozták olyan behatóan, mint a tevékenység-absztrakciót, talán azért, mert egy olyan individuális objektum, mint például a 37-es szám, nem tűnik elég általánosnak ahhoz, hogy az absztrakt tanulmányozás középpontjába kerüljön. Mindazáltal, az objektum-absztrakciók alkotják a *típus-absztrakciók* építőköveit. A típus-absztrakciókat úgy is felfoghatjuk, mint ekvivalencia osztályokat, amelyeknek elemei a $\lambda f[f(x)]$ alakú objektumok, a szokásos alkalmazható műveletekkel és ezeket $\lambda x[\lambda f[f(x)]$-szel jelölhetjük.

A fenti tárgyalás szemlélteti a tevékenység- és az objektum-absztrakciók közötti dualitást. Mutatja, hogy a lambda kalkulus formális absztrakciós műveletével hogyan ragadható meg az absztrakció intuitív fogalma. Ez egyúttal bevezetésül is szolgál az absztrakció még általánosabb tárgyalásához, amelyre a következő pontban kerül sor.

1.3. *Az absztrakció fogalma*

Az absztrakció fogalmát először a görögök tanulmányozták és Platón ideák elméletével szemléltethetjük azt. Platón az olyan fizikai tárgyak eseteit, mint pl. az asztalok, úgy tekintette, mint azoknak az ideális absztrakt objektumoknak a megtestesítőit, amelyek egy ideális világban nagyobb realitással léteznek, mint a nem tökéletes fizikai világban. Az 1/a ábra szemlélteti a kapcsolatot az asztalféleség



1. ábra
Az absztrakt specifikáció és annak esetei közötti összefüggés

Platón-féle absztrakt fogalma és annak a valóságos világbeli esetei között. Az asztalféleség absztrakt fogalma a konkrét eseteknek (asztaloknak) egy ekvivalencia osztályát határozza meg. Az asztalféleséghez hasonló absztrakcióknak a specifikációs problémája — amelynek a nehézségét már a görögök is felismerték — az, hogy találjunk az asztalféleségekre egy olyan adekvált absztrakt jellemzést, amely magába foglalja az asztalok összes eseteit és kizárja az objektumoknak azokat az eseteit, amelyek nem asztalok.

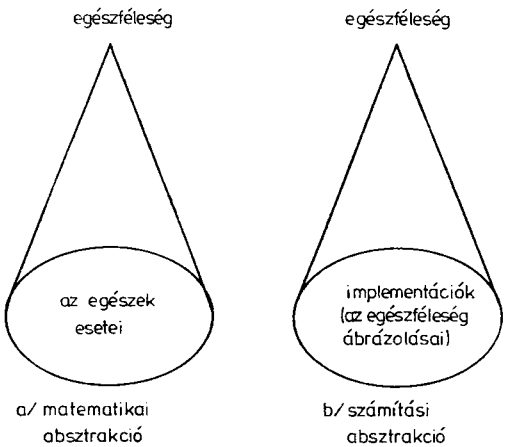
Az 1/b ábra egy absztrakció és az esetek megfelelő ekvivalencia osztálya közötti általános összefüggést illusztrálja. Az absztrakció alapvető gondolata, hogy az esetek (objektumok, helyzetek, folyamatok) ekvivalencia osztályainak „lényeges" viselkedését azok közös jellemzőivel fejezzük ki. A közös jellemzőket úgy is felfoghatjuk, mint az absztrakcióhoz tartozó konkrét esetek ekvivalencia osztályának egy *invariánsát*. Az absztrakció lehetővé teszi, hogy különbséget tegyünk egy invariáns absztrakt specifikáció által meghatározott absztrakt viselkedés tanulmányozása és az absztrakcióhoz tartozó konkrét esetek egyedi viselkedése között. (A hasznos absztrakcióktól megköveteljük hogy az absztrakció (asztalféleség) invariáns jellemző tulajdonságai „egyszerűbbek" legyenek, mint annak esetei (az asztalok). Sajnos, ez nem mindig lehetséges a kiszámítási absztrakciókra (a programok absztrakcióira).) A kiszámításokkal kapcsolatban az absztrakt jellemzők gyakran azt specifikálják, hogy *mit* kell kiszámítani, az esetek pedig azt specifikálják, hogy a számítást *hogyan* kell realizálni.

A matematikában az absztrakció fogalmát a predikátumkalkulussal szemléltethetjük, amely az igazság absztrakt fogalmát definiálja (1/c ábra). A predikátumkalkulusban az axiómák és a következtetési szabályok a jól megfogalmazott formulák (tételek) egy ekvivalencia osztályát definiálják, amely pontosan az igaz formulákból (a tautológiákból) áll.

Az igazság axiomatikus jellemzésében elért siker bátorította a formális rendszerek még általánosabb használatát az absztrakciók specifikálásában. A *Peano- axiómáknak*, mint az egészekre vonatkozó absztrakt specifikációknak a használata, egy másik példája a formális rendszerek használatának a matematikai absztrakciók

specifikációjában. A *Peano- axiómák* az egészféleségeknek a fogalmát absztrakt objektumok (egészek) formájában definiálják, amelyeket egy konkrét reprezentációval (modell, implementáció) kell képviselni, ha azok kiszámítási tulajdonságait kívánjuk tanulmányozni.

A 2/a és 2/b ábrák azt szemléltetik, hogy ugyanahhoz az absztrakcióhoz a konkrét esetek minőségileg különböző ekvivalencia osztályait is rendelhetjük. A 2/a ábra ekvivalencia osztálya absztrakt egészeket, a 2/b ekvivalencia osztály pedig sémákat tartalmaz az egészféleség specifikációjának a reprezentálására (implementálására).



2. ábra
Matematikai és kiszámítási absztrakció

A programozási nyelvi kutatások inkább egy absztrakt viselkedés implementációjának az ekvivalencia osztályaival foglalkoznak, mintsem az absztrakt objektumoknak — amelyek a viselkedés konkrét esetei — ekvivalencia osztályaival. (A 2/a típusú absztrakciókat „elsőrendű absztrakcióknak" nevezhetjük, mivel az absztrakcióból kapott egyedek egy ekvivalencia osztályán kvantálunk, míg a 2/b típusú absztrakciókat „másodrendű absztrakcióknak" nevezhetjük, mivel ott inkább egy eljárás vagy adat-absztrakció realizációi felett kvantálunk, mint az egyedek felett. Ez a szóhasználat a predikátumkalkulussal analóg, ahol az elsőrendű predikátumkalkulus csak az egyedek feletti kvantifikációt teszi lehetővé, míg a másodrendű predikátumkalkulus lehetővé teszi a függvények és a relációk feletti kvantálást is. A másodrendű predikátumkalkulus azonban absztrakt függvények és relációk feletti kvantálást enged meg, ezért az analógia nem pontos.) Ez a különbség az absztrakció matematikai és kiszámítási fogalma közötti különbséget tipikusan jellemzi.

Az összes implementációk ekvivalencia osztályai, amelyek egy adott absztrakt viselkedést (megnyilvánulást) realizálnak, általában igen bonyolultak is lehetnek. Például, az összes rendezési algoritmusok ekvivalencia osztálya (ezeket az algoritmusokat KNUTH részletesen tárgyalja [16] munkájában) nagyon sok különböző rendezési algoritmust tartalmaz, amelyeknek ekvivalenciáját nagyon nehéz bebizonyítani. Annak a problémának az eldöntése, hogy vajon két tetszőleges eljárásnak

ugyanaz-e az absztrakt viselkedése (ugyanazt a függvényt valósítják-e meg) általában eldönthetetlen (még parciálisan is [75]). Így a kapcsolat egy absztrakt eljárásspecifikáció (amelyet egy bemeneti-kimeneti összefüggés határoz meg) és egy program (amely az eljárást megvalósítja) között nem konstruktív, mert ha az lenne, akkor a programok ekvivalenciája eldönthető lenne annak meghatározásával, hogy azok azonos specifikációval rendelkeznek-e.

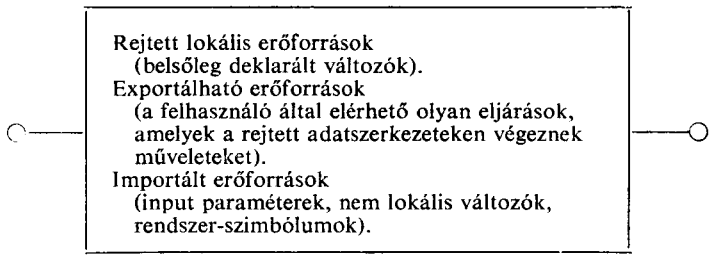
Azokat az absztrakciókat, amelyek a programozási nyelveknél fellépnek, általában *viselkedési* jellemzők határozzák meg, ezért ezeket *viselkedési absztrakcióknak* nevezhetjük. A viselkedési absztrakcióknak két fontos fajtája a következő: 1. az *akciók,* amelyeket operátorokkal és a programozási nyelv eljárásaival lehet ábrázolni, továbbá bemeneti-kimeneti összefüggésekkel lehet absztrakt módon jellemezni; 2. az *objektumok,* amelyeket típusokkal és a programozási nyelvek típusdefiníciós eszközeivel lehet definiálni, továbbá algebrákkal lehet absztrakt módon jellemezni. A nyelv primitív operátorai és típusai az absztrakt viselkedés nyelv által definiált *formái,* míg az eljárás- és adatabsztrakciók az absztrakt viselkedés *programozó által definiált formái,* ezeket szemlélteti a 3. ábra.

	Tevékenység absztrakciók	Objektum absztrakciók
Nyelv által definiáltak:	primitív operátorok értékadás vezérlési struktúrák	adattípusok strukturált típusok
Programozó által definiáltak:	blokkok eljárások	definiált típusok CLU nyalábok (clusterek) Alphard formák monitorok

3. ábra
A programozási nyelvek által támogatott viselkedési absztrakciók

A nyelv által definiált absztrakciókat (operátorokat és típusokat) az *implementációt* végző személy valósítja meg és annak részletei rejtve maradnak (közömbösek) a felhasználó számára, míg a programozó által definiált absztrakciókat a *modulok* valósítják meg, amelyeknek a programozási nyelvi primitívák formájában való megvalósításai a modul felhasználója előtt hasonlóan rejtve maradnak. Az absztrakt viselkedés nyelv vagy rendszer által való támogatásához bármely szinten szükség van egy határfelületre *(interface-re),* amely elválasztja az implementációs környezetet a felhasználói környezettől. Az implementációs környezeten belül vannak lokális (rejtett) erőforrások, amelyek az absztrakt viselkedés implementációs céljára felhasználhatók, de a felhasználó számára nem hozzáférhetők. Lehetnek még „importált" erőforrások is, amelyeket a felhasználó paraméterek útján direkt módon, vagy nem lokális paraméterek által indirekt módon ad meg, lehetnek továbbá „exportált" erőforrások, ezeket a modul szolgáltatja a felhasználónak, s ezekkel kell definiálni a modul absztrakt viselkedését. Mindezt a 4. ábra szemlélteti.

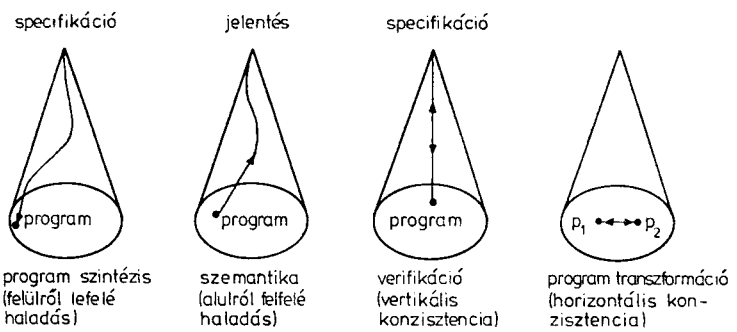
A modul elősegíti az általa megvalósított feladat egy „absztrakt" viselkedési „mit" specifikációjának és e feladat „hogyan" implementációjának a szétválasztását. A modul viselkedésének az implementációtól független, szisztematikus specifikálására való képességének fontos következménye van a programozási módszertanban. Ez a képesség teszi lehetővé a modulok használatára vonatkozó tervezési döntések



4. ábra
„Rejtett" erőforrásokkal rendelkező modulok

megtételét az implementációra való tekintet nélkül. Lehetővé teszi továbbá a modulok implementációjára vonatkozó döntések meghozatalát a rendszer kifejlesztésének olyan pontján, amely a modulok felhasználására vonatkozó döntésektől független. Ez nagymértékben leegyszerűsíti a karbantartást is azáltal, hogy a rendszer módosításának hatását lokalizálja. Lokalizálja a modulok specifikációját és verifikációját is, ezáltal lehetővé teszi, hogy az implementációk („és" specifikációk) helyességét minden egyes modulra függetlenül határozzuk meg és végül nagymértékben leegyszerűsíti a modul módosításakor a verifikáció feladatát.

Az elméleti kutatásoknak négy lehetséges irányát különböztethetjük meg, amelyek az absztrakt specifikációk és a hozzájuk tartozó ekvivalencia osztályok közötti kapcsolatra vonatkoznak, ezeket szemlélteti az 5. ábra.



5. ábra
Az absztrakció szintjei közötti leképezések
(a négy alapeset összehasonlító szemléltetése)

A felülről lefelé haladás *(top-down)* esetében azt a folyamatot vizsgáljuk, amely az absztrakt specifikációt a megfelelő ekvivalencia osztály egy elemévé (programmá) transzformálja, ezt szokták *programszintézisnek* nevezni. Az alulról felfelé haladó *(bottom-up)* esetben a programokhoz társított absztrakt specifikációk meghatározására törekszünk és ezt szoktuk *programozási nyelvi szemantikának* nevezni. A vertikális konzisztencia a program és annak specifikációja között fennálló konzisztencia verifikálását jelenti, és *programverifikáció* néven ismert. A horizontális konzisztencia a programok transzformálásának felel meg, mely transzformációnak

a célja másik, lehetőleg hatékonyabb program előállítása, ugyanazon az ekvivalencia osztályon belül. Ez *programtranszformáció* néven ismert. Azokat a kutatásokat, amelyek ezeken a területeken folynak az anyagban részletesen tárgyaljuk, de a fenti négy kutatási irányt annak illusztrálására mutattuk be itt röviden, hogy a programozási nyelvi kutatások több területe voltaképpen az absztrakt viselkedés specifikációi és implementációi között fennálló kapcsolat valamely aspektusával foglalkozik.

A fenti területeken folyó kutatások nehézségei a következők:

— A *specifikáció problémái* abból a tényből adódnak, hogy egy specifikációs nyelvben egy absztrakt viselkedési specifikáció számos esetben sokkal bonyolultabb és kevésbé szemléletes lehet, mint a programmal való konkrét specifikáció.

— Az *implementáció problémái* ab ól a nehézségből származnak, hogy vertikális vagy horizontális átmeneteket kell létrehoznunk a problémareprezentációk terében, amelyet a fentiekben egy kúppal jelöltünk ki.

A specifikáció problémáiról kiderülhet, hogy azok végső soron alapvetőbbek, mint az implementáció problémái és lehet, hogy éppen a lényegükből fakadóan korlátozzák majd e kutatások eredményeinek gyakorlati alkalmazását.

A specifikációkat műveleti értelemben, vagy absztrakt értelemben definiálhatjuk. A műveleti értelemben vett specifikációkat egy specifikációs modell (reprezentáció, implementáció) és a viselkedési ekvivalencia (izomorfizmus) egy fogalmával adjuk meg, míg az absztrakt specifikációkat definiáló jellemzők (axiómák) specifikációjával adjuk meg. A műveleti specifikációknak számos előnyük van, mivel ezek nyújtanak bizonyos intuíciót és segítséget annak megértéséhez, hogy hogyan lehet egy absztrakciót realizálni, más szempontból azonban alkalmatlanok is lehetnek, mivel a részletek előtérbe állításához és bonyolultsághoz vezetnek, elhomályosítva a tulajdonképpeni absztrakt funkcionális viselkedést és mivel absztrakt célokat, mint amilyen például az érthető, jól strukturált programok előállítása, implementációtól függő technikai célokkal (mint amilyen pl. a go-to utasítások kiküszöbölése) helyettesíthetnek.

A műveleti és az absztrakt specifikációk között fennálló kapcsolatot a 2. fejezetben a blokk struktúrájú nyelvekben fellépő objektumok élettartamával, a 3. fejezetben az adatabsztrakciók specifikációjával, a 4. fejezetben pedig a programozási nyelvek specifikációjával összefüggésben tárgyaljuk.

Az absztrakt és a műveleti specifikáció között fennálló kapcsolat hasonlít ahhoz, amely a felülről lefelé *(top-down)* és az alulról felfelé *(bottom-up)* történő modellalkotás között fennáll. A műveleti specifikációk „alulról felfelé" működnek, a problémamegoldásra szolgáló normálformáknak, mint egy konkrét esetnek a kiragadásával, míg az absztrakt specifikációk felülről lefelé dolgoznak. Számos összefüggésben nincs logikai különbség a végső eredményben, mivel a konkrét eset és egy ekvivalencia reláció kiragadásával kapott ekvivalencia osztály ugyanaz, mint amit a tulajdonságok vagy axiómák egy absztrakt halmazának a kiragadásával kapunk. A két megközelítés között azonban valószínűleg nagy a pszichológiai különbség, és egy jó problémamegoldónak könnyen kell tudni váltania a problémamegoldás folyamán a részben megoldott probléma felülről lefelé, vagy alulról felfelé történő szemléletei között.

Az absztrakció két különálló szubjektív okát különböztethetjük meg:

1. Egy egyedi esetből történő *általánosítás*, a viselkedési jellemzők egy adott halmazával rendelkező összes esetek halmazára.

2. *Egyszerűsítés,* azért mert az absztrakciót leíró tulajdonságok részhalmaza figyelmen kívül hagyja az egyedi esetekhez tartozó lényegtelen részleteket.

A hasznos viselkedési absztrakcióknak egyaránt kell általánosítaniuk és egyszerűsíteniük. Az alkalmazási programozásban, a viselkedési absztrakciókkal kapcsolatos problémák egyike az, hogy a viselkedési absztrakciók általában egyik specifikációs nyelvben sem írhatók le egyszerűen. Tekintve, hogy képtelenek vagyunk az általánosítást és az egyszerűsítést egyidejűleg végrehajtani, ezért az absztrakt specifikációk a gyakorlatban is és az elméletben is csak korlátozottan hasznosíthatók. Ezt a kérdést a 4. fejezetben fogjuk részletesen tárgyalni.

3. A programozási nyelvek fejlődése

2.1. *A programozási nyelvek első generációja* (1950—58).

A programozási nyelvek első generációja abban az időben alakult ki, amikor a számítógép szűkösen rendelkezésre álló eszköz volt és jogosan kétkedve ítélték meg a magas szintű nyelvek gazdaságosságát. Ennek ellenére, ez az időszak rendkívül termékeny volt. 1958-ig sok alapvető programozási nyelvi ötletet kidolgoztak és meg is valósítottak, olyan nyelvek keretén belül, mint a Fortran I., Algol 58, Flowmatic és IPL 5, továbbá kitűzték a jövő számára a nyelvek továbbfejlesztésének általános irányait. A nyelvi fejlesztéseknek négy irányát lehet megkülönböztetni, amelyeknek mindegyike megteremtett egy implementációs technológiát és programozási nyelvi építőelemek egy alapkészletét, az igényesebb célokat kitűző második és harmadik generációs nyelvek számára. Ezek az irányok a következők:

1. A szimbolikus assembly nyelvek a gépi nyelvhez közelálló jelölést nyújtanak, de megteremtik a géptől való függetlenség bizonyos fokát a szimbolikus műveleti és címkódok, a helytől független szubrutinok, továbbá a pszeudo kódok által. Az utóbbiak előremutatnak ahhoz a mechanizmushoz, amellyel még igéretesebb géptől független műveletek definiálhatók. Az 1960-as évek makroassemblerei és makrogenerátorai az 1950-es évek szimbolikus assemblereinek nyelvi technológiáján alapszanak.

2. A tudományos problémák numerikus megoldására szolgáló nyelvek sok számolást és viszonylag kevés adatot igényelnek. A Fortran I., Algol 58 első generációs programozási nyelvek voltak a jobban kicsiszolt Fortran II., és Algol 60 második generációs nyelvek közvetlen elődei.

3. Az adatfeldolgozási problémákra kidolgozott nyelvek — ellentétben a program-intenzív műveletekkel — adat-intenzív műveleteket igényelnek. Ezek a nyelvek megkísérelték összeötvözni a gazdagabb adatleírási lehetőségeket a természetes nyelvi stílusú probléma-specifikációval. A folyamatosan fejlődő technológia, társulva a programozási alapkészlettel, teremtette meg a befogadó környezetet a Cobol számára.

4. Listafeldolgozó nyelvek változó méretű és szerkezetű objektumokkal való műveletek elvégzését segítik elő. Az IPL 5 nyelv felfogható egy hipotetikus számítógép assembly nyelveként, amely listafeldolgozást végez és az IPL 5 kézikönyve pedig a hipotetikus számítógép műveleteinek megfelelő nyelvet definiálja. Az IPL 5 inkább

az IPLx ($x \leqq 5$) nyelvjavaslat-sorozatnak a tetőpontját jelenti, mint a kiindulópontját olyan későbbi nyelvek kidolgozásának, mint a Lisp. Ennek ellenére, az IPL 5 felhasználásával folytatott alapvető mesterséges intelligenciakutatások adták meg az indító lökést a későbbi listafeldolgozó nyelvek kifejlesztéséhez.

2.2. *Második generációs nyelvek* (1959—61)

A négy második generációs nyelv, a Fortran II, Algol 60, Cobol és Lisp, a programozási nyelvek intenzív fejlesztési szakaszának a végtermékét képviseli és ugyanakkor a későbbi programozási nyelvek fejlesztésének egy viszonylag stabil alapvonalát jelöli ki. A tárgyalás során, minden egyes megvizsgálandó nyelv esetében azokat a specifikus tulajdonságokat fogjuk hangsúlyozni, amelyek különösen a későbbi nyelvfejlesztések szempontjából lényegesek. A Fortrannál megvizsgáljuk a részprogramok (főprogram és szubrutinok) közötti kommunikációra kifejlesztett technikát és a közös objektumok használatára szolgáló módszert (a COMMON és EQUIVALENCE utasítások, valamint a formális és aktuális paraméterek megfeleltetésének segítségével). Az Algol 60-nál az objektumok azonosítókhoz való rendelésének a módszerét és a memóriakiosztás alapelveit tárgyaljuk meg. A Cobol esetében az adatleíró lehetőségeket és a természetes nyelven történő programozáshoz való közelítést mutatjuk be. A Lisp-nél megvizsgáljuk az interpretatív nyelvi definíciót, az azonosítók összekapcsolásának technikáját (beleértve a FUNARG technikát), továbbá a listák és szimbolikus kifejezések rekurzív függvényei közötti leképezést.

2.2.1. *Fortran*

A Fortran program egy főprogramból, névvel ellátott szubrutinok egy halmazából és COMMON adatblokkok egy halmazából áll. E két utóbbi halmaz egyike vagy mindegyike esetenként üres is lehet. A szubrutinok és a COMMON adatblokkok nevei globálisan ismertek azért, hogy egy részprogramból bármelyik szubrutint vagy COMMON adatblokkot el lehessen érni. Azok a nevek, amelyek nem szubrutinok vagy COMMON adatblokkok nevei, lokálisak abban az értelemben, hogy ezekhez nem lehet hozzáférni annak a részprogramnak az érvényességi körén kívülről, amelyikben azok felhasználásra kerülnek.

A Fortran nyelv tervezése egy olyan implementációs modellen alapszik, amely megköveteli, hogy a részprogramok és a COMMON adatblokkok helyfoglalási igényei a betöltéskor ismertek legyenek azért, hogy a nevek a lefordított programban relatív címen legyenek ábrázolhatók. Ez leegyszerűsíti mind az implementációt, mind pedig a végrehajtási folyamatok lényegi megértését, ellenben a Fortran adatobjektumokat „rögzített méretűre" korlátozza és megkövetel bizonyos nyelvi sajátosságokat, mint például a dinamikus határokkal rendelkező tömböknek és a rekurzív szubrutinoknak a nyelvből való kizárását. (Ez világosan mutatja, hogy a nyelv tervezőinek szeme előtt lebegő implementációs modell megcsonkítja a felhasználóval való érintkezés funkcionális nyelvi jellemzőit. Az olcsóbb hardware-nek és az implementációs modellek jobb megértésének egyik következménye, hogy lehetővé válik ezeknek a korlátoknak a lazítása.)

A szubrutinok formális paramétereinek a kezdőértékei megadhatók a megfelelő aktuális paraméterekre történő hivatkozással. (A paraméterek érték szerint is át-

adhatók, úgy hogy a szubrutin a paraméterek lokális másolatával dolgozik, azaz más folyamat a szubrutin végrehajtása közben nem tudja azt módosítani. A Fortran szabvány gondosan kizárja a meggondolásból azokat az eseteket, amelyeknél a paraméterátadás kétféle módszere eltérő eredményre vezethet.) Azok az aktuális paraméterek, amelyek egyszerű változók, azt eredményezhetik, hogy az az adatobjektum, amelyet a hívóprogram aktuális paramétere megjelöl, a hívott program formális paraméterével közös lesz. Azok a paraméterek, amelyek összetett kifejezések vagy literálisok a híváskor kiszámítódnak és a megfelelő formális paraméternek értékként átadódnak.

A változók és a memóriarekeszek között a végrehajtás idejére rögzített megfeleltetés nem szükségképpen kölcsönös és egyértelmű. Egy adott részprogramban a különböző nevű változókhoz ugyanazt a memóriarekeszt rendelhetjük az EQUIVALENCE utasítással. A különböző részprogramok változóihoz pedig a COMMON változók útján, vagy a formális és aktuális paraméterek előbb említett megfeleltetésével rendelhetünk közös memóriarekeszeket. A Fortran a program nevei (változói) által a memóriarekeszek és a memóriarekeszekben tárolt adatobjektumok hajlékony közös használatát engedi meg. Ez a közös használat egyrészt forrása a nyelv kiszámítási hatékonyságának, másrészt azonban melegágya a rejtett programozási hibáknak is, amint azt a 4. fejezetben kifejtjük.

2.2.2. *Algol* 60

Az Algol 60 program egy blokkból áll. A blokk a deklarációknak egy halmaza, amelyet az utasításoknak egy sorozata követ. Az utasítások maguk is blokkok lehetnek úgy, hogy a program egy tetszőleges rögzített mélységig egymásba skatulyázott komponens blokkokat tartalmazhat. A deklarációk lehetnek eljárásdeklarációk, amelyek egy eljárás névből, egy formális paraméterspecifikációból és egy ezeket követő eljárástörzsből állnak. Ez az utóbbi blokk is lehet. Az eljárásdeklarációk tetszőleges mélységben tartalmazhatnak egymásba skatulyázott blokkokat és eljárásdeklarációkat.

Az Algol 60 programban használt összes azonosítókat (neveket) abban a blokkban vagy eljárásban kell deklarálni, illetve specifikálni, amelyben a felhasználási pontjuk van. Az azonosító felhasználásának egy-egy esete mindenkor az azonosítónak arra a legbelső deklarációjára vagy specifikációjára korlátozódik, amely a felhasználási pontot magába foglaló blokkban, vagy eljárásban van. Az azonosító deklarációjának vagy specifikációjának az érvényességi köre az összes olyan programpontok halmaza, amelyekben az azonosítót csak az adott azonosítódeklarációval összhangban lehet használni.

Mi úgy fogunk hivatkozni a blokkokra és az eljárásokra, mint programmodulokra, az azonosítók deklarációjára, vagy specifikációjára pedig, mint azonosító-definícióra. A programmodulok rendeltetése a nomenklatura új szintjének a definiálása és a modulon belül definiált azonosítóknak az elrejtése a modult magába foglaló környezet elöl. A programmodulon belül definiált azonosítókat a programmodulra nézve lokálisnak nevezzük.

A blokkokba azáltal léphetünk be, hogy a végrehajtás során átlépjük a statikus programban a modul határát, míg az eljárásba eljárás hívással léphetünk be. A programmodulba való belépés a modul végrehajtásának egy esetét hozza létre és azt eredményezi, hogy a lokális azonosítók az adatobjektumok (lokális adatobjektumok)

újonnan létrehozott eseteire korlátozódnak. A programmodulba való belépéskor létrejött lokális adatobjektumok léte felfogás szerint „örökké" tartó, de azok a modulból való kilépéskor hozzáférhetetlenné válnak és törölhetők. Ezért a lokális adatobjektumokat úgy képzelhetjük el, mint amelyek a programmodulba való belépéskor jönnek létre és kilépéskor megszűnnek.

Az objektumok megvalósításának verem modellje és a hozzátársuló „törlés a belépéskor" modell nagyon hasznos a forrásnyelvi konstrukciók futásközbeni jelentésének a megértésében és az Algol 60 objektumok számára műveleti szemantikát szolgáltat. Mindamellett az összes adatobjektum „eredendő szemantikája" az, hogy az mindaddig létezik, amíg ahhoz hozzá lehet férni. A műveleti „törlés a belépéskor" szemantika származtatott szemantika, amely a lényeges szemantikával ekvivalens, mivel az Algol 60-at gondosan úgy tervezték meg, hogy miután kiléptünk egy programmodulból a modul lokális objektumaihoz való hozzáférhetetlenség biztosítva legyen. Ennek a származtatott műveleti szemantikának az eredendő szemantika státusára való előléptetése két veszélyt rejt magában:

a) A származtatott szemantika nem általánosítható új nyelvi struktúrákra. A „létrehozó — törlő szemantika" olyan nyelvekre való általánosításának a problémáit, amelyek mutatóváltozókat is tartalmaznak, a 4.2. fejezetben tárgyaljuk.

b) Az implementációnak a származtatott műveleti szemantika által meghatározott absztrakt modelljét nem szabad összetéveszteni a tényleges implementációkban használt mechanizmusokkal, amelyek az effektivitást helyezhetik előtérbe a felfogásbeli egyszerűséggel ellentétben. Főleg az optimalizáló Algol 60 fordítóprogramokról van szó, amelyek a blokkok és nem rekurzív eljárások részére a memóriakiosztást a fordítás során végzik el, ezért a futás közbeni memóriakiosztás többletműveletei csak azokban a programokban lépnek fel, amelyek dinamikus nyelvi lehetőségekkel dolgoznak és azok a programok, amelyek nem élnek ezekkel a lehetőségekkel, éppen olyan hatékonyan futnak, mint a megfelelő Fortran programok.

A fenti megfontolás alapján az Algol 60 számára a törlési modell nem eléggé általános ahhoz, hogy bizonyos nyelvi általánosításokat felöleljen és nem eléggé speciális sem ahhoz, hogy a tényleges implementációk jó modellje legyen, mindazonáltal, ez a modell fontos bepillantást nyújt a blokkstruktúrájú nyelvek szemantikájába. A fenti tárgyalás egyaránt mutatja a műveleti modelleknek a hasznosságát és egy speciális műveleti modellhez való túlzott ragaszkodás veszélyét.

Az Algol 60-nak az a megszorítása, amely szerint az adatobjektumok szorosan annak a programnak a végrehajtásához kötődnek, amelyben létrehozták azokat, megfelelő lehet olyan numerikus algoritmusok számára, amelyek a bemenő adatokkal úgy végeznek műveleteket, hogy az előállított eredményben minden részszámolásnak a teljes hatását felölelik, nem megfelelő ez azonban olyan adatokkal végzett intenzív számolásokhoz, amelyeknél az adat „túléli" azokat az algoritmusokat, amelyek műveleteket végeznek vele.

Az adatobjektumoknak egy olyan státuszra való felemelkedése, amely a vele műveletet végző programoktól független, a Fortranban valósul meg a COMMON adatblokkokkal (amelyek embrionális adatbázisok) és a lokális adatobjektumokkal, amelyek a programmodul végrehajtásától függetlenül léteznek. A Cobol és az adatbázis-kezelő nyelvek ezt tökéletesítik olyan adatleírásokkal ellátott file-ok bevezetésével, amelyeknek a hozzáférhetősége nem kötődik a szöveg alapján meghatározott programmodulokhoz.

2.2.3. *Cobol*

A Cobol program a következő részekből áll:

— Azonosító rész, amely a programozót és a programot azonosítja.
— Környezetet leíró rész, amely a hardware konfigurációt, valamint a logikai és fizikai be- és kimenő adatfolyamok közötti kapcsolatot specifikálja.
— Adatleíró rész, amely az adatfile-ok szerkezetét írja le.
— Eljárást leíró rész, amely azokat a folyamatokat definiálja, amelyek az adatfile-okon végeznek műveleteket.

Az azonosításra, számítógép-környezetre, adatkörnyezetre és eljárásspecifikációra való moduláris felosztás, az Algol 60 eljárásmoduljainak és deklarációs részének szintjétől eltérő szintet képvisel. A modularitás kérdéseinek gondos figyelembevétele lehetővé teszi a programozók számára, hogy az adatbázis-tervezés kérdéseivel az adatbázisokon műveleteket végző programoktól elkülönítve foglalkozhassanak. Ez a megoldás jelentősen hozzájárult a Cobolnak, mint adatfeldolgozási nyelvnek a sikeréhez. A Cobol eljárásleíró részének az a hiányossága, hogy nincs eljárásmodulja, elfogadhatónak tűnik, mivel az alkalmazói programok általában inkább adatokra, mint algoritmusokra orientáltak.

A Cobolnak az a kísérlete, hogy természetes nyelvi stílusú programozást vezessen be (például „ADD *A* TO *B* GIVING *C*"-t „$C=A+B$" helyett) nem bizonyult sikeresnek, mivel az ilyen programok jobb olvashatóságát ellensúlyozza a hibák megtalálásának és a programkarbantartásnak nagyobb problémája. Kiemelendő viszont, hogy a Cobol volt az első nyelv, amely lehetővé tette olyan „rekord" adattípusok használatát, amelyek névvel ellátott heterogén típusú komponensekkel és input-output műveletekkel rendelkeznek, továbbá megengedte a teljes rekordra vonatkozó értékadást.

A Cobollal szerzett tapasztalatok azt mutatják, hogy a hardware, a bemeneti-kimeneti csatornák, adatfile-ok stb. szintjén, az alkalmazott programozási környezet gondos figyelembevétele lényeges előfeltétele az adatstruktúrák és az eljárások szintjén egy alacsonyabb szintű modultechnológia hatékony kifejlesztésének.

Felmerült egy olyan általános célú nyelvnek az igénye, amely egyaránt alkalmas algoritmusra és adatra orientált alkalmazásokra, amely összeötvözi annak a lehetőségét, hogy az adatbázisokat a velük műveletet végző programoktól függetlenül írjuk le, egy kifinomult eljárás résszel. Úgy tűnik azonban, hogy a programozási nyelvek és az adatbázisok területén dolgozók a nyelvtervezésnek nem ilyen egységes megközelítése felé haladnak.

2.2.4. *Lisp*

A Lispben kétféle típusú objektum van, a lista és az atom. A Lisp műveletei lehetővé teszik elemek kiválasztását, komponensekből listák felépítését, egyenlőség, valamint atomszerűség eldöntését. Rendelkezik ez a nyelv egy függvényabsztrakciós operátorral („lambda") és egy olyan operátorral („label"), amely a függvényabsztrakciók elnevezését és a rekurzív definíciókat lehetővé teszi. Vezérlési struktúrái között megtalálható a feltételes elágazás és a függvény alkalmazása.

A Lispnél a kiértékelés megértésének a kulcsa a következő: a programok egyöntetű módon leképezhetők listastruktúrákra és a Lisp-programok kiértékelése egy

olyan programmal (értelmezővel) írható le, amely azt specifikálja, hogy a listastruktúrákat reprezentáló programok, argumentumaikkal együtt, hogyan alakíthatók át értékekké. Ez az értelmező egy meglehetősen egyszerű Lisp-programmal írható le, amelyet Lisp „Apply" függvénynek nevezünk [22].

A Lisp program primitív műveletei a hagyományos programozási nyelvek primitív műveleteitől jelentősen különböznek. A „tiszta" Lispben nincs értékadó utasítás és „goto" utasítás. A számításra két fő módszer létezik: paraméterek és függvénynevek értékekhez való kapcsolása és ilyen értékek változókkal való helyettesítése, ahogyan azok a függvénytörzs balról jobbra történő kiértékelése során előfordulnak. A váltóérték megfeleltetések (összekapcsolások) — amelyeket a lambda és a label műveletek hoznak létre — egy ún. „környezeti listára" kerülnek, amely lista „*last-in-first-out*" sorrendben vizsgáltatik meg a változó aktuális értékének a meghatározása céljából, így a változó értékét mindenkor a legutóbbi összekapcsolás határozza meg.

A kiértékelés folyamata hasonló ahhoz a folyamathoz, amellyel a lambda kalkulusredukcióban egy kötött változó összes előfordulásaihoz párhuzamos behelyettesítéssel rendelünk értéket. Igaz viszont, hogy a Lisp behelyettesítési mechanizmusa a változókhoz szekvenciálisan rendel értékeket, abban a sorrendben, amilyen sorrendben azok a függvénytörzs kiértékelése során előfordulnak. Ez — bizonyos esetekben — azt eredményezheti, hogy a függvénytörzsben egy adott szabad változó különböző előfordulásaihoz különböző értékek rendelődnek. Ezért ennek a kiértékelési módnak nagyon sajátos hatása lehet.

A fenti probléma a Lisp funkcionális argumentum (FUNARG) lehetőségével kerülhető el. Ez megengedi, hogy az aktuális paraméterkifejezés szabad változói a környezeti listában olyan értékekhez legyenek kapcsolva, amelyek az összekapcsoláskor voltak érvényesek és nem pedig a felhasználáskor jönnek létre. Ez biztosítja, hogy az aktuális paraméterek szabad változói magukkal vigyék az ő környezetüket és amikor azok felhasználásra kerülnek, a helyes környezetnek megfelelően értékelődjenek ki.

A tiszta Lispnek a szintaxisa egyszerű, egy nagyságrenddel egyszerűbb, mint az Algol 60 szintaxisa és egyszerű szemantikája van, amelyet a Lisp „Apply" függvénye definiál. A Lisp Apply függvény által megvalósított interpretatív nyelvi definíciója mintaként szolgált a későbbi műveleti nyelvi definíciók számára. A *McCarthy-féle leképezés* a listák és a szimbolikus kifejezések (*S*-kifejezések) között, valamint a Lisp-programok és a parciális (*M*-kifejezésekkel ábrázolt) rekurzív függvények között levő leképezés, lehetővé tette a kiszámíthatóság egy matematikai elméletének a kifejlesztését Lisp számítási primitívák formájában. A Lisp nyelv jelentős szerepet játszott alkalmazási nyelvként a mesterséges intelligencia területén és kiindulási pontként szolgált a kiszámíthatóság műveleti és matematikai modelljeivel kapcsolatos elméleti munka számára.

2.3. *Harmadik generációs nyelvek* (1962—69)

2.3.1. *PL/1*

A PL/1 [23] kísérlet a Fortran, Algol és Cobol nyelvekhez hasonló nyelvek „jó" tulajdonságainak egy általános célú nyelv keretében való egyesítésére és számos olyan ígéretes új elképzelés megvalósítására, amely az 1960-as évek elején tűnt fel. Vannak ebben a nyelvben a Fortran szubrutin moduljaihoz hasonló külső eljárások és a külső

eljárásokon belül megtaláljuk az Algolhoz hasonló blokkstruktúrát. A struktúrák Cobolhoz hasonló adatleírási lehetőségeket nyújtanak. A bázisos változók segítségével pedig a Lisphez hasonló nyelvek listafeldolgozó primitivái könnyen szimulálhatók. A kivétel kezelését az ON utasítás támogatja és létezik a nyelvben bizonyos egyszerű multitaszk lehetőség is.

2.3.2. *Algol* 68

Az Algol 68 [24, 25] a kifejező erejét azzal éri el, hogy kisszámú ortogonális fogalomból kiindulva és menet közben minimalizálva a korlátozásokat, lehetővé teszi, hogy ezekből a fogalmakból összetett struktúrákat építsünk fel. Ez egy blokkszerkezetű nyelv, amely a „módok" egy végtelen halmazával nyújt támogatást, ideértve a teljesen tipizált paraméterekkel ellátott eljárásokat, hivatkozásokat (mutatókat) és struktúrákat (rekordokat), valamint az automatikus módkonverzióra (koercióra) szolgáló szabályok kifinomult halmazát. Eszközt nyújt korlátozott élettartamú lokális objektumok, valamint globális élettartammal, de lokális hozzáféréssel rendelkező halmazobjektumok létrehozásához, támogatja továbbá a típusdefiniálás különböző lehetőségeit és a szemaforok szinkronizálásával megvalósított „kollaterális" (párhuzamos) számítást. Az átdolgozott Algol 68 jelentés a nyelvet kétszintű nyelvtan *(Van Wijngaarden nyelvtan)* segítségével definiálja, amellyel meg lehet adni a (nem környezetfüggetlen) szintaktikai deklarációkat és a szemantika félig formális leírásait. Annak, aki meg akar ismerkedni a nyelvvel, a [24] jelentés elolvasása előtt ajánljuk a [25] cikk feldolgozását.

2.3.3. *Snobol* 4.

A Snobol 4 [26] szöveggel végzendő műveletekre kifejlesztett nyelv, amely hatékony mintaillesztő műveletekkel rendelkezik a szöveges és sematikus adattípusok számára. A szövegekkel és a sémákkal végzendő műveleteket beleágyazták egy jól megtervezett általános célú keretbe, amelyhez jól használható eljáráshívó mechanizmus, programozó által definiálható adattípusok és rendszerfejlesztési eszközök (például nyomkövetés) tartoznak. A Snobol 4-nek van egy olyan, viszonylag géptől független makro nyelvű implementációja is, amelyet könnyen át lehet vinni új gépekre.

2.3.4. *Simula* 67.

A Simula 67 [27] egy nagyon jól használható új típusú modult vezetett be, amelyet „*class*" modulnak nevezünk. A *class modulnak* van egy eljárás komponense, amely társrutinként *(coroutine)* hajtható végre, vannak lokálisan deklarált eljárásai és adatszerkezetei. Ezek az eljárások és adatszerkezetek a class modul aktivizálásai között is léteznek és hozzáférhetők. Rendelkezik a nyelv egy „*subclass*" technikával is, amely lehetővé teszi származtatott modulok definiálását. Ezek a származtatott modulok a „*szülő class modulok*" speciális esetei, annak folytán, hogy ezek mindazokkal a tulajdonságokkal rendelkeznek, amelyekkel a szülő modul rendelkezik,

és vannak olyan tulajdonságaik is, amelyeket a „subclass"-ban definiálunk. A class modulban megtestesülő elképzelés erős hatást gyakorolt a modularitással kapcsolatos jelenlegi kutatásokra. (Lásd a 3.4. pontot.) A subclass technika segítséget nyújt a fogalmak hierarchikus definíciójához, [27] azáltal, hogy megengedi a subclass modul fogalmi beágyazását abba a környezetbe, amelyet a szülő class modul határoz meg, anélkül, hogy ezt a beágyazást fizikailag is meg kellene valósítani. Ezt a módszert hierarchikus objektumok modellizálásának alapjaként felhasználták a Smalltalk [40] programozási rendszerben.

2.3.5. *Pascal*

A Pascal [28] olyan Algol-szerű nyelv, amely az adattípusok nagyon gazdag választékával rendelkezik, ideértve a „range" típusokat, struktúrákat és korlátozott mutatókat („pointer" változókat), a gondosan megtervezett adatdefiníciós lehetőségeket. A nyelv tervezésekor az egyik cél az általános felhasználhatóság volt, de ugyanakkor ezt a nyelvet az egyszerűség, hatékonyság és megbízhatóság jellemzi. Ez a nyelv jelentős szerepet játszott a programhelyesség bizonyításával kapcsolatos kutatásokban, mivel ez volt az első axiomatikusan definiált nyelv [85]. Kiindulópont volt ez a nyelv az 1970-es évek bizonyos moduláris, és általános célú nyelvei számára és első számú alapját képezte a DOD1 fejlesztéseknek [4]. Van azonban a Pascal nyelv tervezésének néhány kisebb-nagyobb hiányossága, amelyek közül néhányat a [29] és a [34] munka tárgyal.

2.3.6. *APL*

Az APL nyelv [30] a tömbökre vonatkozó műveletek gazdagságával tűnik ki. Ezek a műveletek szükségtelenné teszik a nyelvi szinten végrehajtott iterációt a legtöbb tömbre vonatkozó művelet esetében, megteremtve annak a feltételét, hogy a felhasználó a tömbelemek helyett magukat a tömböket tekintse primitív objektumoknak. Adatdeklarációs lehetőségei nincsenek, vezérlési struktúrái nagyon kezdetlegesek. Úgy tűnik, hogy gyakran olyan agyafúrt programok írására ösztönöz, amelyek bonyolult feladatokat a hagyományostól eltérő, meglepően tömör módon írnak le, szokatlanul primitív műveletekkel. Az ilyen programokat azonban nehéz olvasni, megérteni és gyakran a hatékonyságuk is elmarad a velük azonos feladatot megoldó hagyományos programok hatékonysága mögött. Az APL nyelvet kitűnő interaktív rendszer támogatja. Ez jelentősen hozzájárult ahhoz, hogy a nyelv a tudományos feladatokat megoldó programozók és más felhasználók között népszerűvé vált. (Az előbbi típusú felhasználók a számítógépet asztali számítógép módjára használják, rendszerint csoportosan.) Az APL-t, mivel lehetővé teszi a tömbök elemi objektumokként való kezelését, igen magas szintű nyelvnek tekinthetjük.

2.3.7. *Basic*

A Basic [31] nagyon egyszerű párbeszédes nyelv, amelyet széles körűen használnak a középiskolákban és más intézményekben, a kezdő programozók első programozási nyelveként. Úgy tűnik, hogy a strukturált programozással ellentétben, bátorítja az explicit címkék és a „goto" utasítások használatát a programozás-

ban. Az a jelenlegi nézet, hogy a programozónak a problémát strukturált szerkezetben kell megfogalmaznia, azt sugallja, hogy a Basic nyelvnek első nyelvként való megtanulása rossz programozási szokások beidegződéséhez vezethet és ezért veszélyes. Meg kell azonban említeni, hogy a Basic nyelvet nagyon széles körűen használják, és ez a nyelv az 1960-as évek egyik fő programozási nyelvévé vált. Más, az itt említett programozási nyelvekkel ellentétben úgy tűnik, hogy a Basic nyelv nem tartalmaz semmi olyan jelentős fogalmat, amely lényeges lehetne a jövő programozási nyelveinek tervezése szempontjából.

3. Nyelvi fogalmak

A nyelvi fogalmak tárgyalása azoknak a nyelvi módszereknek az azonosítását célozza, amelyek lehetővé teszik, hogy a programozó a kiszámíthatóság szempontjából hasznos formában fejezze ki az absztrakt viselkedési formákat. Mi a típusokat, a vezérlési struktúrákat és a modulokat választottuk ki, mint ilyen célt szolgáló három nyelvi módszert. A típusok közös műveletekkel ellátott objektumok részhalmazainak a megnyilvánulási formáját azonosítják. Ilyenek például az egészek, vagy a közös hozzáférési tulajdonságokkal rendelkező objektumok részhalmazai, a tömbök. A vezérlési struktúrák meghatározzák a tevékenységi sorozatok viselkedési sémáit. A modul egy olyan módszer, amely egy összetett megnyilvánulási forma implemetációjának a beburkolására szolgál oly módon, hogy azt az implementációra vonatkozó ismeretek nélkül lehesen felhasználni. Így tehát a típusok és a vezérlési struktúrák az objektum, illetve a tevékenység megnyilvánulásának absztrakt formáit határozzák meg, míg a modulok az összetett megnyilvánulás absztrakt formáinak a definiálását támogatják.

3.1. *Típusok*

A nyelv típusai a nyelv objektumainak a halmazát egyforma megnyilvánulási formával rendelkező részhalmazokra osztják fel. A nyelv olyan egyszerű típusai, mint az *integer, real* és *Boolean* a közös matematikai műveleteknek alávetett megnyilvánulások egyöntetűségét tükrözik. A matematikusok behatóan tanulmányozták az adattípusok algebrai tulajdonságait, de a végtelen sok elemű halmazzal rendelkező adattípusok esetében — ilyen például az egészek és a valósak halmaza — a gépi ábrázolás véges volta miatt levágási hatások lépnek fel, amelyet még nem tudunk adekvát módon matematikailag jellemezni.

A típusoknak egy jelentős osztályát képezik az ún. strukturált típusok, amelyek a hozzáférési műveleteknek alávetett megnyilvánulás egyöntetűségét képviselik. HOARE [32] a strukturálási mechanizmusokat a következőképpen osztályozta:

Descartes-szorzat (amely a rekordok definiálására használható).

Discriminált unió (amely a típusok diszjunkt egységeinek definiálására szolgál).

Tömb (amely egy olyan leképezés, amelynek az értelmezési tartománya az index tartomány és értékkészletét a tömb elemeinek típusa adja).

Hatványhalmaz (amely egy típus elemeiből képezett részhalmazok halmaza).

Sorozat (amely lehet szöveg, lista, file és más dinamikus adatszerkezet modell).

A programozási nyelvekben általánosan támogatott típusok a megnyilvánulás olyan egységes formáit testesítik meg, amelyek egyrészt hasznosnak bizonyultak a problémamegoldásban, másrészt hatékonyan ábrázolhatók a számítógépeken is.

A típusdeklarációk lehetővé teszik, hogy a programozó leszűkítse a változókhoz rendelt értékek tartományát. Az objektumokra vonatkozó típusspecifikáció és a változókra vonatkozó típus deklaráció függetlensége az APL és a Snobol 4 nyelv példájával szemléltethető. Ezekben a nyelvekben megvan az objektumokra vonatkozó típusoknak egy jól kifejlesztett fogalma, de ugyanakkor megengedik azt is, hogy a végrehajtás különböző pontjain különböző típusú objektumokra hivatkozzunk változókkal. (Ezeket a nyelveket meg kell különböztetni a „típusnélküli" nyelvektől, ilyen például a lambdakalkulus, amelyben az összes objektum egyszerű típusú [33], vagy a gépi nyelvektől, amelyek nem nyújtanak az adatobjektumok számára semmiféle típusvédelmet.) A típusok objektumainak ábrázolásai diszjunktak. Ez lehetővé teszi a végrehajtás során annak az ellenőrzését, hogy egy operátor operandusai mindenkor a megengedett típusúak-e? Például, a Snobol 4 objektumai magukkal cipelik a típuskódot a változók értékábrázolásának részeként. Természetesen az explicit típusok és a típuskódok ellenőrzése hely és időtöbbletet eredményez, bár ezt azokban a nyelvekben el lehet kerülni, amelyekben a változók típusait a fordítás során határozzuk meg. Ez a többlet azonban elfogadható egy olyan nyelvben, mint a Snobol 4, amely komplex primitív műveletekkel rendelkezik és amelynek a végrehajtási ideje csak nagyon csekély mértékben növekszik a típuskód-ellenőrzés miatt.

A programozási nyelvet nagyon gondosan kell megtervezni ahhoz, hogy az az összes deklarált változóra adekvát típusspecifikációt szolgáltasson. A nyelvet akkor nevezzük *erősen típusos nyelvnek,* ha minden változónak a típusa a fordítási idő alatt meghatározható. Az Algol 60 nem erősen típusos nyelv, mivel a név szerint hívott paramétereknek a típusspecifikációját nem kötelező megadni, vagy mivel a típus eljárások eljárásparamétereinek teljes típusát nem szükséges specifikálni. A PL/1 nyelv sem erősen típusos nyelv, mivel a mutató *(pointer)* értékű változók esetében nem követeljük meg, hogy azoknak a változóknak a típusai specifikálva legyenek, amelyekre a pointerek mutatnak. A Pascal nyelv sem erősen típusos nyelv, mivel nem követeli meg, hogy azoknak a paramétereknek a típusa specifikálva legyen, amelyek eljárások. Hiányzik továbbá ebből a nyelvből egy jól definiált típus-ekvivalencia fogalom és problémák vannak a változó rekordokkal kapcsolatban is, (erre még később vissza fogunk térni).

Az Algol 68-at az erősen típusosság elérése céljából gondosan tervezték meg, ennek ellenére a típusekvivalencia fogalma inkább szintaktikai, mint szemantikai fogalom. A programozó által definiált két típus akkor tekinthető ekvivalensnek, ha azok azonos módon építhetők fel primitív típusokból. A „mode polar = [1:2] real;" és a „mode complex = [1:2] real;" típusdeklarációk, amelyeket a polar x; complex y deklarációk követnek, azt eredményezik, hogy x és y azonos típusú lesz annak ellenére, hogy a polar és a complex változókon értelmezett műveletek eléggé különbözőek. (Vegyük észre, hogy a Pascal nyelv nem definiálja, hogy a primitív típusokból azonos módon felépített típusok vajon ekvivalenseknek tekintendők-e.) Ez azt illusztrálja, hogy a típus fogalma, mint az alkalmazható műveletekkel meghatározott megnyilvánulási mód, nem mindig kompatibilis a típusnak azzal a fogalmával, amely egy összetömörített objektum komponenseire vonatkozó egységes hozzáférési

mechanizmussal fejezhető ki. Ahhoz, hogy elkerüljük egy típusnak, mint amilyen például a polar, átkonvertálását egy valóságos adattípussá, műveleteket kell értelmeznünk a típuson és el kell rejtenünk az adatszerkezetet ([1:2] real) egy olyan beburkoló definícióban, amelyben a hozzáférési műveletek csak magára az adatszerkezetre és nem pedig annak komponenseire vonatkoznak. Ilyen beburkolási technikának a kifejlesztése képezi a tárgyát a programozási nyelvekkel kapcsolatos kutatásoknak, ahogyan azt a CLU és az Alphard [43, 44] esete is mutatja.

A kétféle típusszemlélet között, tudniillik a megnyilvánulási módok és a hozzáférési struktúrák különböző szemlélete között rejlő ellentétet nagyon világosan szemléltetik az APL-beli tömbök, amelyek annak ellenére, hogy számos összetett tömbművelettel rendelkeznek, mégsem tekinthetők összetömörített adattípusoknak, mivel a tömb elemekhez az összetett tömbműveletek keretén kívül is hozzá lehet férni. Az az APL programozó viszont, aki lemond az elemekre vonatkozó műveletek használatáról és csak az összetett tömbműveletekkel dolgozik, a tömböket úgy kezeli, mint adattípusokat. Az egyes elemekhez való hozzáférés lehetőségét feladva, a programozó arra kényszerül, hogy a tömböket olyan absztrakt objektumoknak tekintse, amelyeknek megnyilvánulását az összetett műveletek határozzák meg. Természetesen vannak olyan esetek, amelyekben a tömb elemeihez való közvetlen hozzáférés hasznos és természetes. Ez rávilágít arra a tényre, hogy különböző környezetek, amelyekben egy adott objektum megjelenik, az absztrakció különböző szintjét követelik meg.

A Pascal nyelv scalar és subrange típusai egy másik olyan példáját szolgáltatják a típusoknak, amelyek problémákat okoznak, mivel ezek az adattípusnak csak néhány és nem pedig az összes tulajdonságával rendelkeznek. A véges scalar típusoknak, ilyen például {hétfő, kedd, szerda, csütörtök, péntek} és a subrange típusoknak, például „hétfő, ..., csütörtök", van egy rendezettségük, amelyet elemeiken a hozzájuk rendelt „megelőző" és „követő" függvények határoznak meg. Azokra a problémákra, amelyeket a zártságnak ez a hiánya okoz, HABERMAN [34] mutatott rá.

Az egész és a valós adattípusokra vonatkozó műveletek zártságának a hiánya miatt fellépő problémákkal a túlcsordulási mechanizmus foglalkozik. A tömb indexhatárai a fordítóprogramoknak jelentenek speciális feladatot, mivel a megengedett műveletekre ezek nem zártak, függetlenül attól, hogy az indexhatárokat típusként, vagy nem típusként kezeljük. Ha a subrange típusokat nem tekintjük típusoknak, akkor lehetővé válik, hogy azokat a levágási problémákat, amelyek a műveletek nem zárt volta miatt adódnak, a nyelv szintjén vegyük tekintetbe. A scalar és a subrange típusok hasznos absztrakciók, mivel számos szövegkörnyezetben az objektumok véges halmazáról mint absztrakt dolgokról beszélünk, de a műveletekre nézve fennálló zártság hiánya azt jelenti, hogy az absztrakció megnyilvánulásában anomáliák vannak jelen és ezekkel gondosan kell bánni.

Az erősen típusos nyelvekben az a megszorítás, hogy a változók csak egyszerű típusú objektumokra hivatkozhatnak, esetenként kényelmetlen lehet. Például, a Pascal nyelv a különböző indexhatárokkal rendelkező tömböket (pl. array [1:20], array [2:21]) különböző típusúaknak tekinti. Mivel az eljárások argumentumai rögzített típusúak kell, hogy legyenek, ebből következik, hogy egy 20 elemű tömb rendezésére szolgáló eljárás nem tud dolgozni egy 21 elemű tömbbel. Ennek a problémának egyik lehetséges megoldása az, hogy az összes egydimenziós tömböket azonos típusúaknak tekintjük. Ez azonban egy másik szövegkörnyezetben okozhat proble-

mát. Ilyen például az „$a:=b$" értékadás, ahol a és b egydimenziós tömbök. Jobb megközelítést kapunk, ha felismerjük, hogy a változóknak egyszerű típusú objektumokra való korlátozása ebben az esetben teherré válik, és megengedjük ennek a megszorításnak a feloldását nyelvi szinten. Megfelelő módszer a paraméterrel rendelkező (polimorf) típus bevezetése, amely az összes olyan típusoknak az egyesítését jelöli, amelyek a típusparaméter speciális értékeihez tartoznak.

A változók erős típusossága, társulva azzal a követelménnyel, hogy az operátorok csak egyszerű adattípusú értékeket vehetnek fel, olyan erős korlátozás, amely bizonyos esetekben mesterkélt, nehezen olvasható és rossz hatékonyságú programokhoz vezet. Ezért az erősen típusos nyelvek általában tartalmaznak egy olyan lehetőséget, amely megengedi a programozónak, hogy visszalépjen az erős típus szemléletétől, amikor az már inkább terhes, mintsem előnyös. Az ilyen lehetőségek különböző változatait GESCHKE, MORRIS és SATHERTHWAITE [35] munkája tárgyalja.

3.2. *Vezérlési struktúrák*

A legalapvetőbb vezérlési struktúra a goto utasítás. A vezérlésátadás legtöbb esete azonban logikai folyamat részeként fordul elő. Ilyen például az iteráció esete, amely a goto utasítással társult vezérlési művelet egyszerű vezérlésátadását egy vezérlési séma nagyobb vezérlésátadásába ágyazza bele. A jó programozási gyakorlat azt mutatja, hogy a programozónak fel kell ismernie a logikai vezérlési sémákat az algoritmusokban és olyan magasabb szintű vezérlési sémákat kell használnia, amelyek a goto utasításnál megfelelőbbek.

Az a felismerés, hogy a goto utasítások kritikátlan alkalmazása szükségtelenül bonyolult programokat eredményez, vezetett el az 1960-as években ahhoz a mozgalomhoz, amely a goto utasításokat kívánta száműzni a programozási nyelvekből [37]. A Bliss [38] nyelv a sikeres goto nélküli nyelvek egyik példánya, amely a *Digital Equipment Corporation* hivatalos rendszerprogramozási nyelvévé vált. A goto utasítás azonban természetes vezérlési primitíva, amelyből más vezérlési struktúrák építhetők fel. KNUTH [36] meggyőzően mutatott rá, hogy a goto utasításnak a használata számos esetben indokolt.

Az esetek elemzésével és az iterációval társuló vezérlési sémák „if", ill. „while" utasításokkal fejezhetők ki. A vezérlési sémák matematikailag vonzók, mivel az esetek elemzése és az induktív okfejtés a matematikai okoskodás legfontosabb formái közé tartozik és ezek ügyesen automatizálhatók [80]. Ezek a struktúrák egy belépéssel és egy kilépéssel rendelkeznek és ez a tulajdonságuk felhasználhatóvá teszi ezeket a lépésenkénti finomítás folyamatában, amikor egy olyan élet, amely azt specifikálja, hogy mit kell kiszámítani, helyettesítünk egy részletesebb specifikációval, amely már azt specifikálja, hogy hogyan kell az eredményt előállítani. BÖHM és JACOPINI [39] kimutatta, hogy az „if" és a „while" utasítás, a blokk utasítással együtt a primitíváknak egy olyan teljes rendszerét alkotja, amellyel minden kiszámítható függvény kifejezhető.

A fenti előnyök ellenére azonban az „if", „while" és a „blokk" vezérlési séma nem elégséges a gyakorlati programozás számára, mivel vannak olyan gyakran előforduló vezérlési sémák, mint például a ciklusból történő rendellenes kilépés, amelyek nem képezhetők le ezekre a vezérlési sémákra anélkül, hogy ne vezessünk be szükségtelen logikai változókra vonatkozó újabb műveleteket és le ne romboljuk

a program természetes struktúráját. BÖHM és JACOPINI eredménye ugyanabba az osztályba tartozik, mint az az eredmény, hogy minden programot meg lehet valósítani a *Turing gép* primitíváinak a segítségével. A vezérlési struktúrákkal kapcsolatos munkák egyik tárgya az alkalmazói programokban általánosan előforduló vezérlési struktúrák azonosítása és olyan nyelvi szintű struktúrák nyújtása, amelyek ezeket a sémákat megvalósítják.

Az „if" utasításnak az az általánosítása, hogy tetszőleges számú alternatíva közül lehessen választani, vezetett el a „case" utasításhoz. A

$$\underline{case}\ i\ \underline{in}\ a_1, a_2, \ldots, a_n\ \underline{out}\ \alpha\ \underline{esac}$$

utasítás $1 \leqq i \leqq n$ esetén az a_i tevékenység, különben pedig az α tevékenység végrehajtását eredményezi.

A „case" utasítás egy másik változata:

$$\underline{case}\ var\ \underline{in}\ val_1 : a_1,\ val_2 : a_2, \ldots, val_n : a_n\ \underline{out}\ \alpha\ \underline{esac},$$

amely $\underline{var} = \underline{val_i} (1 \leqq i \leqq n)$ esetén az a_i különben pedig az α végrehajtását eredményezi.

A kiválasztási mechanizmus további általánosításai vezettek el a

$$(p_1 : a_1, p_2 : a_2, \ldots, p_n : a_n)$$

formához, amely annak az a_i tevékenységnek a végrehajtását eredményezi, amelyre a p_i állítás igaz. Ez a konstrukció az alapja a mesterséges intelligenciában előforduló alkalmazások vezérlési struktúráinak [68], és a nem determinisztikus *Dijkstra-féle „garded command"* szerkezetnek [60]. Ezekkel a 4.2.4. pontban foglalkozunk.

A „while" utasítás egyik hiányosságát mutatja a következő ügyetlen konstrukció, amely kezdő programozási tanfolyamokon gyakran felbukkan:

```
{olvasd be és készítsd elő az adatot}
while jó az adat do
{számolj az adattal, olvasd be és készítsd elő a következő adatot}
```

Ezt a konstrukciót egy *„loop-while-repeat"* vezérlési szerkezettel kerülhetjük el (lásd KNUTH [36] munkáját), amely megragadja az iteráció természetes szerkezetét, azt, hogy minden iterációnak van egy indítási fázisa, amelyet a befejeződés-vizsgálat követ és végül a számítási fázis zár le:

```
loop
  {olvasd be és készítsd elő az adatot}
while jó az adat do
  {számolj az adattal}
repeat
```

Ha az indítási fázist elhagyjuk, a hagyományos „while" ciklust kapjuk.

A ciklusból történő, szokásostól eltérő kilépés problémája kapcsolatban van az eljárásból szokásostól eltérő módon való kilépéssel (lásd a 4.3.4. pontot). A ciklusból való többszöri kilépésre egy szisztematikus mechanizmust javasol a [37]

munka, és ez a Bliss-nyelv esetében meg is valósult. ZAHN [36] egy olyan általános vezérlési struktúrát javasolt, amely megengedi a ciklus magjából a többszörös kilépést úgy, hogy azok „eseményjeleket" adnak, de az eseményekhez tartozó befejeződési tevékenységeket összesűríti egy belépéssel és egy kilépéssel rendelkező modulra.

loop until ⟨esemény$_1$⟩ or ... or ⟨esemény$_n$⟩ in

⟨ciklus törzs⟩

repeat

then ⟨esemény$_1$⟩:⟨befejező tevékenység$_1$⟩

⋮

⟨esemény$_n$⟩:⟨befejező tevékenység$_n$⟩

end loop

ZAHN elgondolása olyan általános, hogy a gyakorlatban fellépő legtöbb vezérlési struktúra ennek speciális eseteként definiálható. Például a

loop számolj test számolj repeat

struktúra úgy modellezhető, hogy a test részét egy befejeződési eseménnyel helyettetítjük, amelyhez befejező tevékenységként az egyszerű kilépés társul. Az ilyen általános vezérlési struktúra azonban nem nyújt „természetes" jelölést azokra a speciális vezérlési struktúrákra, amelyek a gyakorlatban gyakran előfordulnak és nem sok jót ígérne azok helyettesítése vele.

A for konstrukció speciális vezérlési struktúra, amely egy vezérlő változó egymás utáni értékeire szervezett iteráció fogalmát foglalja magába. Ennek a szemantikáját részletesen tárgyalja HOARE [83]. A

$$\text{for } x := e_1 \text{ step } e_2 \text{ to } e_3 \text{ do } S$$

és a

$$x := e_1;\ \text{while } x < e_3 \text{ do } \{S;\ x := x + e_2\}$$

struktúrák közötti kapcsolat világosan mutatja, hogy a for utasítás egy vezérlő változó bevezetésével, továbbá az iteráció és a befejeződés vizsgálata közötti környezetmódosítás speciális tételével specializálja a while utasítást. Az Algol 60-ban, az ilyen iterációs folyamatok lényeges egyszerűségének kompromisszumaként, megengedik az S utasításon belül az x, e_2, e_3 módosítását is, helyt adva a következőkhöz hasonló anomáliáknak:

$$\text{for } x := 1 \text{ step } 1 \text{ until } n \text{ do } \{\ldots\ n := n + 1\}$$

vagy

$$\text{for } x := 1 \text{ step } 1 \text{ until } n \text{ do } \{\ldots\ x := x - 1\}.$$

Ezek és más hasonló problémák elkerülhetők, ha megtiltjuk az x, e_1, e_2, e_3 változtatását az S-en belül és az x változót a for ciklus lokális változójának tekintjük, amelynek az értéke kilépéskor nincs definiálva. Ezt a szemantikát foglalja magába

a következő séma, amely egy lokális környezetet hoz létre a while utasítás végrehajtása számára:

```
begin integer x, incr, final;
      x: e1;
      incr:= e2;
      final:= e3;
      while x final do {S: x:= x+incr}
end
```

3.3. *Vezérlésátadás a modulhatárokon keresztül*

Az itt következő tárgyalás céljából a modult úgy képzeljük el, hogy az egy modultörzsből (utasítássorozatból) és a lokális, valamint a nem lokális elérhető objektumok hozzáférési környezetéből áll. A különböző végrehajtások számára az előbbi közös, az utóbbi pedig esetenként változó. A vezérlés átadása a modulhatáron keresztül nemcsak a kód egy új szakaszára való ugrást jelenti, hanem belépést, illetve kilépést is jelent egy új hozzáférési környezetből. Ilyen vezérlésátadás esetén négy logikailag különböző fajta tevékenység fordulhat elő:

1. Létrehozása (vagy törlése) egy olyan hozzáférési környezetnek, amely a modul végrehajtása egy esetének felel meg.
2. Egy elérési környezet inicializálása.
3. Belépés (vagy kilépés) egy modulba, hívó, visszavevő, vagy visszatérő műveletekkel.
4. Belépés egy modulba egy lokálisan deklarált kívülről ismert eljárásba való belépéssel.

Az eljárás-hívás egy új hozzáférési környezet létrehozását egyesíti a formális és aktuális paraméterek egymáshoz rendelésének inicializálásával, valamint a vezérlés átadásával, amely az új hozzáférési környezetben az eljárás első végrehajtandó utasítására adja át a vezérlést. Megtörténik egy implicit visszatérési címkeparaméter átadása az új hozzáférési környezetnek és ez használható fel a vezérlésnek a hívási pontra történő visszaadásához, egyben törölve az újonnan létrehozott hozzáférési környezetet.

A társrutinok (corutinok) abban különböznek az eljárásoktól, hogy azok hozzáférési környezete a modul végrehajtásától függetlenül létezhet. A hozzáférési környezet létrehozása és inicializálása a belépéstől és a kilépéstől függetlenül is végrehajtható. A társrutin rendelkezik egy olyan saját állapotszóval, amely vezérlésátadás esetén a modulon kívül tárolódik és egy visszavevő parancs hatására kerül onnan vissza a társrutinba való visszalépéskor.

Társrutinokkal lehet szimulálni az egymással kommunikáló párhuzamos folyamatok halmazát. Ilyenkor minden egyes folyamat számára egy-egy társrutint hozunk létre, és a párhuzamos folyamatokból álló halmaz végrehajtásának egy pontját a társrutinok állapotszavainak egy halmazával ábrázoljuk. A valóságos párhuzamos végrehajtást közbeékelt végrehajtásokkal szimuláljuk oly módon, hogy az a párhuzamos folyamatok szinkronizációs feltételeivel konzisztens legyen.

A Simula „*class*" konstrukciója támogatja a class több esetének az egymástól elkülönített létrehozását és a *class törzs* ilyen eseteinek társrutinként való végrehajtását. Ezen túlmenően, a *class konstrukciók* olyan eljárás- és adatdeklarációkat is tartalmazhatnak, amelyeket a környezetből el lehet érni. Valójában a *class konstrukciónak* egy olyan triviális törzse van, amely csak egyszer, a *class* létrehozásának az időpontjában hajtódik végre és a *class*-hoz, az inicializáláson kívüli minden más hozzáférés csak eljárás, vagy adatváltozó segítségével valósul meg.

A Simula 67 *class* mechanizmusának egyik gyengesége, hogy nem rendelkezik a lokális változók hozzáférési módjának a kontrollálására alkalmas hozzáférési kontroll-mechanizmussal. A Simula 67 megengedi, hogy a *class modul* összes változóihoz automatikusan hozzáférjünk, és nincs olyan a *class konstrukció* által meghatározott absztrakciót beburkoló eszköze, amely a lokális adatszerkezeteket elrejthetné. (A Simulában csak a közelmúltban teremtették meg a *class struktura* tulajdonságainak elrejtésére alkalmas lehetőséget. A rejtett tulajdonságú *class konstrukciók* megteremtése a konkurrens Pascal [44] és a Smalltalk nyelvben a jövőben valósul meg.) Absztrakciók megvalósításainak a beburkolására szolgáló hozzáférési kontroll mechanizmusok kulcsszerepet játszanak az olyan moduláris nyelvekben, mint a CLU és az Alphard.

3.4. *Modularitás, beburkolás, adatabsztrakció*

A típusdefiníciós lehetőségek lehetővé teszik, hogy a programozó a megnyilvánulás új formáit definiálja a megnyilvánulás primitív formáinak a felhasználásával. Ezek a lehetőségek potenciálisan nagyon hatékony eszközök a nyelv kifejezőerejének a növelésére. Ezt az 1960-as években ismerték fel és ez a felismerés vezetett el egyrészt a típusdefiníciós lehetőségeknek az Algol 68 és a Pascal-szerű nyelvekbe való beépítéséhez, másrészt a Simula nyelv „class" fogalmához. Ezek a lehetőségek az eljárás, társrutin és az adattípus megnyilvánulás egyesítését jelentik.

A beburkolás fogalma (információelrejtés [44]), egy eddig hiányzó eszközt szolgáltatott, amely jelentősen megnövelte a típusdefiníciós mechanizmusok erejét és gyakorlati hasznát azáltal, hogy megengedi rejtett implementációval rendelkező absztrakt megnyilvánulás új formáinak definiálását a nyelvben. A beburkolásnak az olyan korai formái, amilyenek az Algol 60 blokkjai és eljárásai a belső megnyilvánulást rejtették el a felhasználó elől. A beburkolással kapcsolatos jelenlegi kutatás a következő vonatkozásokban általánosítja az Algol 60 blokk és eljárás mechanizmusát:

1. Dolgoznak az érintkezési felület *(interface)* vezérlési mechanizmusainak a kifejlesztésén, amelyek több rugalmas kommunikációt engednek meg a modulhatáron keresztül, ideértve az átadható tulajdonságokat, paramétereket, az átvett változókat stb.
2. Beburkolási technikákat fejlesztenek ki az absztrakt megnyilvánulás új formái számára, ideértve az absztrakciókat és a konkurrens feldolgozási absztrakciókat (lásd monitorok).

A blokkokra és eljárásokra szolgáló tevékenységre orientált beburkolási technikákat az 1970-es évek elején terjesztették ki objektumokra. A [46] munka a Simula *class struktura* és az adatabsztrakció egy újabb keletű tárgyalását tartalmazza.

A monitorok [49] voltak az elsőnek létrehozott objektum-beburkolási mechanizmusok. Talán azért, mert a konkurrens folyamatok közötti kritikus tartományok megosztásának a vezérlése olyan világosan definiált probléma, amely ezt a fajta megoldást kívánta. Az absztrakt adattípus fogalma, amely 1974-ben jelent meg először az irodalomban [42], általános modellt szolgáltatott absztrakt objektumok beburkolására és elvezetett azokhoz a kutatásokhoz, amelyek egyaránt vonatkoznak az adatabsztrakcióra szolgáló nyelvi támogatást nyújtó mechanizmusokra [43, 44], specifikációra és verifikációs technikákra [47].

Azok a nyelvi kutatások, amelyek az adatok beburkolásának mechanizmusait segítik, az Alphard [43] és a CLU [44] nyelv példájával szemléltethetők. Az Alphardban az adatok beburkolására szolgáló modulok neve *form*, a CLU nyelvben pedig *cluster*. Az Alphard és a CLU a nyelvtervezésnek egymástól nagyon eltérő filozófiáit képviselik. Az Alphard nyelv az adatok beburkolásának a mechanizmusát blokkszerkezetű vázba ágyazza be. Ez „verem nyelv" abban az értelemben, hogy minden objektum, ideértve a „*form*" modulok eseteit is, lokális élettartammal rendelkezik, amelyet annak a programnak az élettartama határoz meg, amelyben az objektum létrejött. A CLU nyelv pedig olyan nyelv, amelyben az összes objektum, a cluster struktúrák eseteit is ideértve, mindaddig létezik, amíg az elérhető.

Az Alphardnak az objektum létrehozásra és értékadásra szolgáló mechanizmusai különböznek a CLU mechanizmusaitól. Az Alphard nyelv deklaráció szerinti objektum létrehozást támogat, amely a programmodulba való belépéskor hajtódik végre, a CLU nyelv pedig egy külön művelet által kezdeményezett objektumlétrehozást támogat. Az Alphard nyelv az értékadást ugyanúgy másolással oldja meg, mint az objektumokra vonatkozó normális értékadó mechanizmusok. Ezzel szemben a CLU az értékadást megosztással *(sharing)* oldja meg (lásd a 4.2.3. pontot).

Az Alphardban a veremre-orientáltság a rögzített méretű adatobjektumok esetében hatékony annyiban, hogy feleslegessé teszi a mutatókat, de különleges kezelést igényel olyan változó méretű objektumok esetén, amilyenek a vermek. Az Alphardban követett megközelítési mód egy másik, alapvetőbb problémája abból a tényből fakad, hogy a lokális élettartamra vonatkozó követelmény a létrehozott objektumnak az „absztraktságát" lerontja, rátéve arra egy ide nem illő lokális élettartam tulajdonságot, amely az objektumok moduláris függetlenségét lerombolhatja azáltal, hogy idő előtt eltünteti azokat.

3.5. *Adatabsztrakciók specifikációja*

Az Alphard nyelvvel kapcsolatos kutatások a nyelvi, a specifikációs és a verifikációs mechanizmusok párhuzamos kifejlesztését húzták alá. Az adatabsztrakciókat az Alphard nyelvben az implementációs modell formájában műveleti hatásukkal specifikáljuk. Például, a vermeket műveletileg azokkal a sorozatokkal adjuk meg, amelyek a „*first*", „*last*", „*leader*", „*length*" primitív műveletekhez tartoznak.

Annak igazolása, hogy a vermek egy adott (mondjuk tömbökkel történő) megvalósítása a specifikációnak egy realizációja, lényegében annak a kimutatásából áll, hogy a megvalósítás konkrét adatstruktúráján értelmezett konkrét műveletek az absztrakt adatstruktúrán értelmezett megfelelő absztrakt műveletek realizációi.

Az adatabsztrakciókra vonatkozó specifikációk lehetnek műveletiek vagy axiomatikusak. Például a vermekre vonatkozó axiomatikus specifikáció a következőképpen adható meg:

— Felsoroljuk a vermekre vonatkozó összes művelet értelmezési tartományát és értékkészletét.

— Megadjuk a műveletek közötti összefüggéseket meghatározó egyenletek (axiómák) halmazát.

Például:

Tartományok:

S: a vermek tartománya,
E: a veremelemek tartománya,
B: a logikai értékek kételemű tartománya,
U: a „nem definiált elem" egyelemű tartománya.

Műveletek:

$create: \rightarrow S,$
$push: \quad S \times E \rightarrow S,$
$pop \quad S \rightarrow S \times U,$
$top: \quad S \rightarrow E \times U,$
$empty?: S \rightarrow B.$

Egyenletek (axiómák):

$empty?\ (create) = true,$
$empty?\ (push\ (s, e)) = false,$
$pop\ (push\ (s, e)) = s,$
$top\ ((push\ (s, e)) = e,$
$pop\ (create) =$ nem definiált elem,
$top\ (create) =$ nem definiált elem.

A vermeknek ez a specifikációja absztrakt specifikáció, mivel mindenféle speciális adatábrázolástól (például tömbök, lineáris listák, sorozatok) független. A [88] munkában szereplő terminológia szerint a verem adattípust többfajtájú algebraként *(multisorted algebra)* adtuk meg, amely egy $M = (S, O, E)$ hármas, ahol S a műveletek értelmezési tartományainak és értékkészleteinek halmaza, O a műveletek halmaza, E pedig a műveletek közötti kapcsolatokat meghatározó egyenletek halmaza.

Bár a fenti axiomatikus specifikáció független az ábrázolástól, azért valójában a verem egy speciális ábrázolását sugalmazza a műveletek egymásutániságával. Például az $x_3x_2x_1$ háromelemű vermet a következő műveleteknek megfelelően ábrázolhatjuk:

$$push\ (x_3,\ push\ (x_2,\ push\ (x_1,\ create))),$$

amely a vermet a létrehozó műveletek történeti sorrendjének formájában specifikálja. Ez a megközelítés azért alkalmazható, mert

a) Az objektumokat egyértelműen meghatározza az azokat előállító konstrukciós (pl. *create, push*) műveletek sorozata.

b) Minden objektum előállítható ilyen műveletekből képezett sorozattal.

c) Mivel az objektumok az azokat felépítő műveletek sorozatával ábrázolhatók, a művelet megnyilvánulási tulajdonsága teljesen definiálható azokkal az axiómákkal,

amelyek azt specifikálják, hogy a műveleteket hogyan alakítják át a műveletek egymás utáni sorozatát.

A *pop (push (e, s))=s* axiómát úgy tekinthetjük, mint a „*pop*" művelet olyan definícióját, amelyet egy korábban végrehajtott „*push*" művelettel adunk meg. A specifikáció szintjén a műveleteknek ez az egymástól való függése bonyolulttá teszi a verifikációt, mivel a műveletekkel kapcsolatos bizonyításokat nem lehet egymástól függetlenül végrehajtani. Az adatabsztrakciókat általában nem tudjuk felbontani olyan moduláris komponensekre, amelyek egy-egy műveletnek felelnek meg és ezt tükrözik az algebrai specifikáció egyenletaxiómái.

3.6. *A konkurrens (párhuzamos) feldolgozás absztrakciója*

A processzorok költsége a számítási rendszer teljes költségének jelentéktelen részévé vált. Ez az, amiért a multiprocesszálás és az osztott számítógép architektúrák egyre jobban általánossá válnak. A programozási nyelvek és az operációs rendszerek az ilyen számítógép-architektúráknak számos részletét eltakarhatják a felhasználó elől, mindazonáltal vannak bizonyos — a konkurrens és megosztott feldolgozásra vonatkozó — absztrakciók, amelyeket a felhasználónak látnia kell.

Az alapvető absztrakció a *folyamat*. A folyamatok egymással közös adatok vagy üzenetek révén kommunikálhatnak és szinkronizációs műveleteknek vannak alárendelve, amelyek egyrészt az alkalmazás által kirótt sorrendi korlátozásokból, másrészt a számítógépi erőforrásokból adódó *ütemezési* korlátozásokból származnak.

A szemaforok [48] a primitíváknak egy alacsony szintű halmazát szolgáltatják a szinkronizációs feladatok megoldására és az adatok közös használatának vezérlésére. A monitirok [50] már hozzáférési eljárásokkal ellátott, közösen használt adatok beburkolására szolgáló magasabb szintű mechanizmusok, amelyeket beépített szinkronizációs műveletekkel felszerelt adatabsztrakciók egy speciális osztályaként foghatunk fel. A monitorok csak egy folyamatnak engedik meg egyidejűleg az adatokhoz való hozzáférést, és támogatást nyújtanak a végrehajtásra várakozó folyamatok egy belső készenléti sorának. Egy feldolgozás alatt levő folyamat vagy a befejezésig fut, vagy végrehajt egy „késleltető" utasítást, amelynek hatására egy belső monitor sorba áll és várakozik egy olyan „életrekeltő" eseményre, amelynek hatására visszakerül a készenléti sorba.

A konkurrens Pascal [49] a monitor elképzelést valósítja meg egy Pascal-szerű nyelv keretében. Támogatja a párhuzamosan végrehajtható *folyamatokat* és a programozó által definiált, *class*-nak nevezett modulokat. Ezeket a modulokat csak abból a folyamatból lehet elérni és végrehajtani, amelyiken belül deklarálták azokat. Ezek lényegében olyan Simula class konstrukciók, amelyek beburkolt adatdeklarációval és monitornak nevezett közös adatmodulokkal vannak ellátva. Ezeket létre lehet hozni egy olyan környezetben, amely több folyamatból elérhető, paraméterként át lehet adni folyamatoknak, valamint a folyamaton belül „*class*" konstrukcióknak és csak „kölcsönös kizárási módban" lehet ezeket végrehajtani egyszerre egy folyamat által.

A többprocesszoros környezetben futó párhuzamos programok sebezhető pontját képezik a nem reprodukálható hibák, amelyek az időtől függő programviselkedés miatt adódnak. A monitorok, ha megfelelő szinkronizációs védelmekkel együtt

használják azokat, biztosítják a párhuzamos programok viselkedésének a reprodukálhatóságát. A monitorok használata azonban súlyos és néha szükségtelen többletterhet róhat a közösen használt adatobjektumokra. Az adatabsztrakcióknak egy kevésbé korlátozó „*class*" konstrukcióját az [52] munkában vezették be, amelyet „*shared data class*" névvel illettek.

A monitorokkal és más párhuzamos programozási absztrakciókkal kapcsolatos kutatás nemcsak a nyelvek fejlesztése szempontjából jelentős, hanem a moduláris mikroprocesszor architektúrák létrehozása szempontjából is. A jövő számítógépei valószínűleg olyan mikroprocesszorok hálózatai lesznek, amelyek rendelkeznek saját memóriával, feldolgozási képességgel és közösen használt memóriaterületekkel. Az ilyen hálózatban az egyedi mikroprocesszorok funkciójuk szerint konkurrens feldolgozási absztrakciókként foghatók fel, és ezeknek jól megfelel a monitorszerű architektúra.

A monitorok úgy valósítják meg a szinkronizációt, hogy korlátozzák a folyamatok végrehajtását, amikor azok a közösen használt adatobjektumokhoz kívánnak hozzáférni. A szinkronizációnak az ún. *path kifejezésekkel* [55] megvalósított kiegészítő megoldása az, amely explicit módon korlátozza azt a sorrendet, amely szerint a „*taskok*" egy sorozata végrehajtódik. A monitorok természetes megoldásnak tűnnek a kölcsönösen kizáró erőforrás kiosztási probléma esetében, amely nem érzékeny az erőforrásokhoz való hozzáférés sorrendjére. A path kifejezések viszont az alkalmazások által meghatározott szekvenciális korlátozások kifejezésére nyújtanak egy természetes utat. A jövő programozási nyelveinek a párhuzamos folyamatok végrehajtási sorrendjének korlátozására szolgáló mindkét fajta nyelvi mechanizmussal rendelkezniük kell.

A szemaforok, monitorok és path kifejezések megfelelőek az olyan „erősen összekapcsolt" feldolgozó rendszerek számára, amelyekben a folyamatok és rendszerint a processzorok egy közös memórián osztoznak. A megosztott számítási rendszerek és adatfolyam-modellek, a felhasználó által vezérelt párhuzamos folyamatok céljára, a nyelvi primitiváknak a fentiektől teljesen különböző osztályát igénylik. Erre mutatott rá az [54] cikk.

4. Kutatási irányok

4.1. *Bevezetés*

A programozási nyelvek területének aktuális kutatási irányait jól tükrözik a „*Language Design for Reliable Software*" című konferenciára [1] benyújtott előadások, amelyek a nyelvtervezés szempontjait hangsúlyozzák, és a programozási fogalmak formális leírásával foglalkozó munkakonferenciára benyújtott cikkek [2], amelyek az elvi és az elméleti kérdésekre irányítják a figyelmet.

A nyelvtervezési kérdésekhez tartoznak a következők: *objektummodellezés* (a kiszámítható objektumok alapvető absztrakt modelljei); az *összekapcsolás* (olyan technikák, mint értékadás, deklaráció és a változókat értékükkel összekapcsoló paraméterátadás); a *típusdefiníció és a beburkolás* (amelyek azzal foglalkoznak, hogy milyen fajta absztrakt megjelenési formákat kell egy nyelvben támogatni); és a *vezérlés* (a szekvenciális, nemdeterminisztikus, illetve multiprogramozott vezérlésű rendszerek vezérlési formái).

Az elméleti kutatási területekhez tartozik mind a programozási nyelvi konstrukciók, mind pedig az egyes programok *formális specifikációja;* a *verifikáció* (a programok és absztrakt specifikációjuk konzisztenciája); a *program transzformáció* és *programszintézis* (amely lehetővé teszi, hogy „navigáljunk" egy absztrakt megjelenési forma összes ekvivalens megnyilvánulásainak terében) és végül a *denotációs,* a *műveleti* és az *axiomatikus szemantikák,* (amelyek alternatív lehetőségeket nyújtanak ahhoz, hogy a programozási nyelv programjaihoz jelentést társítsunk). A fenti területeken a kutatást támogató, fejlesztés alatt álló matematikai eszközök felölelik a programokkal kapcsolatos okfejtésre szolgáló logikai rendszereket [98], a típusok absztrakt megnyilvánulásának a megragadására szolgáló absztrakt algebrákat [88] és a programok szemantikai jelentésének felismerésére szolgáló rácselméleti modelleket [116].

A nyelvtervezési témáknak általában van egy elméleti oldala, amely a specifikációval és a verifikálással foglalkozik, egy alkalmazásra irányuló oldala, amely a gyakorlati alkalmazhatóságot vizsgálja és egy implementációs oldala, amely az implementáció hatékonyságával és szerkezetével foglalkozik. Ily módon az elméleti és a gyakorlati kutatási témák számos esetben nagyon szorosan összefonódnak.

4.2. *Nyelvtervezési témák*

4.2.1. *Objektummodellezési témák*

Az objektummodellezési témák az objektumok típustól független viselkedésével kapcsolatosak. Ide tartoznak az objektum létrehozására, deklarálására, elérésére és az értékadásra vonatkozó fejezetek éppen úgy, mint az implementáció alapjául szolgáló modellel kapcsolatos kutatások. Ebben a fejezetben a következő objektummodellezési kutatásokat tekintjük át:

a) Objektumok közös használata, mellékhatások,
b) Adatáramlás és megosztott számítási modellek,
c) Ismeretreprezentációs nyelvek,
d) Applikációs modellek,
e) Oldalhatás vezérlése beburkolással.

Osztozás és mellékhatások. A jelenlegi magas szintű nyelvekben különbséget teszünk értékek és olyan módosítható objektumok között, amelyek megtartják azonosságukat, miközben az értékük változik. A módosítható objektumok a felújítható memóriarekeszek absztrakciói, amelyek hozzáférési és értékadási műveletekkel társulnak. Ez a gépi szintű nyelvben meglevő hozzáférés és értékadás megfelelő modellje.

Az objektumokra azonosítókkal hivatkozunk és az objektumoknak lehetnek komponenseik, amelyek más objektumokra hivatkoznak. Azt mondjuk, hogy egy objektum elérhető egy azonosítótól kiindulva, ha létezik a hivatkozásoknak egy láncolata, egy elérési út, az azonosítótól az objektumig. Ha ugyanaz az objektum két azonosítótól kiindulva is elérhető, akkor azt mondjuk, hogy azok az *objektumot közösen használják.*

Ha egy olyan objektumot, amelyet két azonosító közösen használ, módosítunk az azonosítók valamelyikétől odavezető úton, akkor az hatással van arra az értékre,

amelyet a másik azonosító „lát". Ezt a hatást nevezzük *mellékhatásnak*. A mellékhatás egy hatékony számítási technika, amely hatékony eszköz arra, hogy egy kiszámított eredmény a felhasználási helyeken rendelkezésre álljon, de lerombolhatja a modularitást azáltal, hogy megengedi a számítások hatásainak tovaterjedését nem kontrollált módon és ez kényes programozási hibákat okozhat.

Az objektummodellezési kutatások egy fontos osztálya foglalkozik a mellékhatások kontrollálásával és kiküszöbölésével. A mellékhatások részben kontrollálhatók oly módon, hogy a hagyományos *(Neumann-féle)* számítógép modellen belül korlátozzuk a hivatkozások műveletét, például megtiltjuk az „álnévvel" való elérést. (Álnévvel való elérés fordul elő például amikor egy $p(x, y)$ eljárást, amelyben két különböző nevű formális paraméter van, amelyek hivatkozások, azonos aktuális paraméter értékekkel hívunk meg, például: $p(a, a)$. Az álnévvel történő elérés nem mindig ellenőrizhető a fordítás idején, mivel például a $p(a[i], a[j])$ hívás jogossága az i és j változók futásközbeni értékétől függ. REYNOLDS [55] kimutatja, hogy az álnévvel történő hivatkozás egy sokkal általánosabb jelenség speciális esete, amelyet ő interferenciának nevez és javasol néhány szintaktikai korlátozást, amely kiküszöböli a nem kívánatos interferenciát anélkül, hogy eltörölné a konstruktív interferenciát, amely a programmodulok közötti kommunikációhoz szükséges.)

Egy másik sokkal radikálisabb megoldás egy teljesen új számítási modell bevezetése, amelyben egy osztott módon használt módosítható objektum esetében az értékadás fogalmát az objektumok adatáramlásának vagy üzenettovábbításának a fogalmával helyettesítjük, ahol is azok a felhasználási helyekre továbbítódnak.

Adatáramlás és megosztott számítási modellek. A számítás adatáramlási modelljei [54] a vezérlési struktúrának és az objektumoknak egy olyan modelljén alapszanak, amely alapvetően különbözik a hagyományos *(Neumann-féle)* számítógépek modelljétől. A „memóriarekesz-objektumok" fogalmát, amelyhez másolás útján történő, a régi érték megsemmisítésével járó, értékadás és hozzáférés társul, az objektumfolyamok fogalmával helyettesítjük. Az objektumfolyamok a számítás egyik helyéről a másikra áramlanak, és ezekbe a folyamokba az objektumok belépnetnek, illetve kiléphetnek érkezési sorrendben *(first in first out)*. A műveletek soros végrehajtásának fogalma helyébe azok megosztott végrehajtásának fogalma lép, valahányszor rendelkezésre állnak azok az operandusok, amelyeken az operátorok végrehajthatók. Mivel a művelet végrehajtásának egyedüli hatása az, hogy kiemelünk operandusokat a bemenő folyamból és az eredményeket behelyezzük a kimenő folyamba, a mellékhatások nem léphetnek fel.

Az adatfolyam modell nemcsak azért tetszetős, mert kiküszöböli az oldalhatásokat, hanem azért is, mert sokkal közvetlenebb modellt szolgáltat több, a valós életben fellépő alkalmazásra, mint a *Neumann-modell*. Azonban nagyon lassú az előrehaladás az olyan számítógépek és nyelvek kifejlesztésében, amelyek közvetlenül támogatják az adatfolyam számítást. Jelenleg nem világos még, hogy vajon vannak-e alapvető problémák az adatfolyam modellben, vagy vajon további kutatások eredményeként előáll-e egy elfogadhatóan hatékony mellékhatástól mentes általános célú adatfolyam számítógép.

Amikor az adatfolyam rendszernek a színterei lényeges számítási egységek, az adatfolyam rendszer egy osztott számítógép-rendszerré válik. Az adatobjektumokat, amelyek az osztott számítógép-rendszer színterei között áramlanak, üzeneteknek nevezzük. Az osztott rendszerek a kommunikációs kutatási témáknak egy új

halmazát vetik fel, ide tartoznak a műveleti pont és a felhasználási pont közötti adatmozgatási problémák és azok a kutatások, amelyek az osztott adatbázisban több példányban jelenlevő adatobjektumok felújítási kérdéseivel foglalkoznak [58]. Ezek a problémák nem nyelvtervezési problémák, de azért az adatfolyam számításokra szolgáló nyelvek és technikák tervezéséhez tartozó kérdések.

Tudásreprezentációs nyelvek. Az adatfolyam modellek nagyon fontos kutatási területet jelentenek, mivel kifejlesztésük indítéka nemcsak magukba a számítógéprendszerekbe, hanem az alkalmazásokba is visszanyúlik. A tudásreprezentációs nyelvek [74] sokkal természetesebb módon modellezhetők adatfolyam modellekkel, mint a *Neumann-féle* modellel. Az ismeretet az ilyen nyelvekben az adatabsztrakciók *(frames, scripts)* körül kell megszervezni, amelyeknek képeseknek kell lenniük parciális ismeret és több felhasználós szempontok ábrázolására. Az ismert prototípusokkal ellentétben, alakfelismerésen alapuló asszociatív visszakeresést, valamint az ismert prototípusoktól való különbségek alapján álló új jelenségek leírására alkalmas technikákat kell szolgáltatni. A gondolkodási folyamatokat természetesen több aktív folyamat írja le. Az, hogy az alapvető mechanizmusokat nem ismerjük, hajlékony vezérlési struktúrákkal, adatábrázolással, beütemezéssel és erőforráskiosztással rendelkező nyelvet követel meg.

Az ezen a területen dolgozó kutatócsoportok arra törekszenek, hogy lényegében hasonló fogalmakra *(frames, scripts, data abstraction)* saját terminológiát hozzanak létre. Az idő eléggé megérett most már ezeknek a fogalmaknak a szintetizálására és egy olyan ismeretreprezentációs rendszer kifejlesztésére, amely egyesíti a következő kutatási területek eredményeit: mesterséges intelligencia — felhasználva a *frame* terminológiát —, nyelvészet — felhasználva a *script* terminológiát —, programozási nyelvek — felhasználva az adatabsztrakció terminológiáját —, és architektúra kutatások — felhasználva az adatfolyam-számítógépek és a megosztott feldolgozás terminológiáját.

Applikációs nyelvek. Az olyan applikációs nyelvek, amilyen a tiszta Lisp és a lambda kalkulus, megkerülik a módosítható objektumokat, a megosztást és a mellékhatásokat azáltal, hogy behelyettesítik (bemásolják) az objektumokat az azonosító minden előfordulásának a helyére. Az ilyen behelyettesíthető másolásnak a implementációját a jelenleg rendelkezésre álló számítási eszközeink nem támogatják. Szimulált behelyettesítést a Lisp-beli FUNARG [22] segítségével, vagy a statikus és dinamikus láncokon alapuló implementációs modellel valósíthatunk meg. A Lisp 1.5-höz hasonló programozási nyelvek, amelyek támogatják az applikációs számítást, általában rendelkeznek egy kiegészítő „*imperatív*" számítási mechanizmussal — amilyen a Lisp „*replace*" utasítása —, amelynek révén az oldalhatások levezethetők. A nem módosítható objektumokkal történő nagyszámú számítási tapasztalat a Lisp-ben azt mutatja, hogy a programozásnak ez a stílusa hasznos olyan objektumokon végzett műveleteknél, amelyek belső dinamikus memória követelményekkel rendelkeznek, de a módosítható objektumok teljes száműzetése a programozási nyelvekből jelenleg nem tűnik praktikusnak. Az applikációs programozás egy módszeres tárgyalását [14]-ben találhatja meg az olvasó.

A mellékhatások elleni védekezés beburkolással. Az előre nem látott mellékhatások által okozott problémák a nem típussal dolgozó nyelvekben a legsúlyosabbak, amelyekben a műveletek nem értelmezett („mezítelen") adatokhoz is hozzáférhetnek.

A típussal dolgozó nyelvek az adatobjektumok köré a műveleteknek egy védőrétegét helyezik, amelyek megvédik azokat az önkényes hozzáférésektől. A típusok előírhatják, hogy az objektumok ne legyenek módosíthatók (felújító utasításokat nem engedve meg) vagy csak olyan nagyon védett felújítást engednek meg, amely a felújítás előtt kérdéshez és az engedély megadásához van kötve. Az ilyen védelem elég súlyos többletterhelést jelent és általában eltekint az olyan egyszerű adattípusoktól, mint a tömbök. Monitorok az adattípusoknak olyan példái, amelyek súlyos hozzáférési többletteherrel rendelkeznek. Ezek a párhuzamos programok miatt fellépő oldalhatásokkal szemben nyújtanak védelmet, biztosítva az adatszerkezeteik számára a kölcsönösen kizárt hozzáférést.

4.2.2. *Összekapcsolási kérdések*

Az összekapcsolási problémák és objektumok és az azonosítók közötti összefüggésekkel foglalkozó objektummodellezési problémák egy speciális osztályának tekinthetők, és a következőket ölelik fel:

a) Értékadás másolással és közös használattal;
b) Paraméter másolás és osztozás;
c) Élettartam, tárfelosztás és szemantika;
d) Deklaráció, objektum létrehozása ellen;
e) Összekapcsolás fordítás közben, erősen típusos nyelvekben.

Értékadás másolással és közös használattal. Az olyan nyelvekben, mint az Algol 60, Algol 68 és Pascal, az „$x:=y$"-szerű értékadások esetében annak az objektumnak egy példánya, amelyre y-nal hivatkozunk, bemásolódik az x-szel társított módosítható objektum (absztrakt memóriarekesz) helyére. Az értékadási utasítások olyanok, mint „*setg*(x, y)" a Lisp-hez hasonló nyelvekben általában azt eredményezik, hogy x közvetlenül hozzáláncolódik ahhoz az objektumhoz, amelyre y-nal hivatkoztunk és nem pedig azt, hogy x egy olyan módosítható objektumhoz láncolódik, amely az y értékének egy másolatát tartalmazza. Az értékadásnak ezt a formáját *közös használattal történő értékadásnak* nevezhetjük, mivel azt eredményezi, hogy x osztozik a kifejezés bal oldala által meghatározott értéken minden olyan más változóval, amely már hivatkozott az objektumra. Ez ellentétbe állítható a másolással történő értékadással, amely a jobb oldal által meghatározott értéknek az x-szel társított módosítható objektumba való bemásolását eredményezi. A közös használattal történő értékadás nagymértékben csökkenti a programban a módosítható objektumok számát és bizonyos esetekben meg is szünteti azokat. A végrehajtás hagyományos modelljében a közös használattal történő értékadás literális implementációjához szükség van egy további indirekt elérési hivatkozásra. Számos esetben ezt a fordítóprogram segítségével ki lehet küszöbölni. Az összekapcsolás fogalma a Lisp-ben, párosítva a közös használattal, elvezethet a számításnak egy mélyebb felfogású modelljéhez, mint amilyen az összekapcsolásnak és a hagyományos blokkszerkezetű nyelvek értékadásának a sokkal inkább gépre orientált modellje. További kutatásra van szükség annak az eldöntéséhez, hogy az átállás a számításnak erre a modelljére kívánatos-e, praktikus-e a jövő általános célú alkalmazási nyelveiben.

Paramétermásolás és közös használat. A másolással és közös használattal történő értékadásnak megvan a megfelelője a paraméter-összekapcsolás területén is.

Az érték szerinti hívás annak felel meg, hogy a formális paramétert összekapcsoljuk az aktuális paraméter egy újonnan létrehozott másolatával. A hivatkozás szerinti hívás annak felel meg, hogy közös használatúvá tesszük az aktuális és a formális paramétereknek megfelelő aktuális paraméter objektumot. A név szerinti hívás pedig az összekapcsolás egy harmadik formájának felel meg, amelyet behelyettesítéssel történő összekapcsolásnak lehetne nevezni és a $\equiv$ jellel jelölni. A behelyettesítéssel történő összekapcsolás inkább matematikai, mint számítástechnikai fogalom. Ez egy alapvető módszer, amelyet a formális paraméternek az aktuális paraméterhez történő hozzákapcsolására a lambda kalkulusban használnak.

Élettartam, tárfelosztás és szemantika. Az objektumok élettartamának és hozzáférhetőségének nyelvdefiníciós fogalma, valamint az élettartam és a hozzáférhetőségi specifikációk hatékony megvalósítására szolgáló, az implementáció által definiált tárfelosztási stratégiák között fennálló kapcsolat nagyon fontos nyelvdefiníciós kérdés. Fortran, Algol 60 és Algol 68 az objektumok élettartamának és hozzáférhetőségének három különböző filozófiáját szemlélteti, amelyek mindegyike az implementáció egy-egy „kanonikus" modelljének feleltethető meg.

A Fortran programok az objektumok egy rögzített (statikus) halmazán végeznek műveleteket, amelyek nem paraméter azonosítókra vonatkozó összekapcsoláson keresztül elérhetők és a paraméter azonosítók között megoszthatók, felhasználva az érték szerinti hívás módszerét. Futás közben tárfelosztási módszerre nincs szükség.

Az Algol 60 valójában létrehoz „lokális objektumokat" és összekapcsolja azokat „lokális azonosítókkal" a blokkba vagy procedúrába való belépéskor és elvágja ezeket a kapcsolatokat kilépéskor. A lokálisan létrehozott objektumok valójában továbbra is léteznek, de elérhetetlenné válnak, miután az őket létrehozó modul végrehajtásának egy esete befejeződött és ezért azok „törölhetők" a program szemantikájának megváltoztatása nélkül. Felhasználható egy „törlő" tárkiosztási stratégia is [56] és verem segítségével megvalósítható a lokális objektumok létrehozása, valamint törlése.

Az Algol 68-ban vannak „lokális" objektumok, amelyeknek hozzáférhetősége az Algol 60 szabályai szerint vezéreltetik, de vannak „olyan" objektumok is *(heap objektumok)*, amelyek korlátozott módon elérhetőkké válnak egy azonosító számára, ugyanakkor hozzáférhetetlenekké is válhatnak egymást követő azonosító összekapcsolások miatt. A lokális objektumokhoz hasonlóan a *„heap"* objektumok is lényegében tovább léteznek, de csak akkor lehet azokat törölni a program szemantikájának a változtatása nélkül, amikor azok hozzáférhetetlenekké váltak. A hozzáférhetetlenség időpontját azonban a statikus program alapján nem lehet meghatározni. A hozzáférhetetlen objektumok törlésére szolgáló tárkiosztási módszerek „hulladék" gyűjtésre vagy referencia számlálásra szolgáló módszerek, amelyek a programok számára a tárkiosztás hatékonyságának a növelését szolgálják, de nincsenek hatással (vagy legalábbis nem kell, hogy hatással legyenek) a program szemantikájára.

Az Algol 60 és Algol 68 nyelvi struktúra gondosan azokra a lokális objektumokra korlátozódik, amelyek mindig elérhetetlenekké válnak az őket létrehozó modulból való kilépéskor. A kilépés utáni hozzáférhetetlenségnek ezt a tulajdonságát szüntetik meg a PL/1-szerű nyelvek, amelyek lehetővé teszik a modul lokális objektumainak a megosztását olyan nem lokális azonosítók segítségével, amelyeknek élettartama hosszabb, mint azoké a lokális adatoké, amelyekre hivatkoznak. A lokális

adatok törlési stratégiák szerinti automatikus törlése, ilyen esetben törölt adatokra való hamis hivatkozást idézhet elő. Az ilyen hamis hivatkozások nemcsak hogy kényes programhibák lehetséges forrásai, de olyan szemantikai anomáliák is, amelyek bizonyos fokig megkötik kezünket olyan objektumtulajdonságok definiálásában, amelyekben az objektumok felhasználói megbízhatnak.

A „*loggó hivatkozások*" problémája [57] rávilágít annak a fontosságára, hogy a nyelvi implementáció kérdéseitől el kell választani a nyelvi szemantika belső kérdéseit. Az a népszerű gondolat, hogy a blokkstruktúrájú nyelvek objektumai a blokkba való belépéskor jönnek létre és a kilépéskor törlődnek, egy példa arra, hogyan lehet az implementáció fogalmát a szemantikai alapelvekre úgy irányítani, hogy közben elhomályosodik az a tény, hogy a törlés egyszerűen egy mechanizmus a hatékony implementációhoz és nem pedig az objektumok belső tulajdonsága. Az objektumok élettartamának fontos alapelve, hogy az objektumok folyamatosan léteznek, ha már egyszer azok létrejöttek, az az alapelv pedig, hogy az el nem érhető objektumok a program szemantikájának a megváltozása nélkül törölhetők, a szemantikával nem rendelkező programtranszformációs szabálynak egy példája (lásd 4.3. szekciót).

A legtöbb jelenlegi programozási nyelv struktúráját részben már meghatározza az, hogy mit lehet hatékonyan implementálni és ez lesz — sőt kell is, hogy ez legyen — a helyzet a jövőben is. Az Algol 68-hoz hasonló „*stack-heap*" nyelvek és a Lisp-hez, Clu-hoz hasonló „*pure-heap*" nyelvek elfogadását részben az fogja meghatározni, hogy mennyire vagyunk képesek hatékony modelleket kifejleszteni ilyen nyelvek implementálására. Az implementáció „*stack-heap*" modelljeivel kapcsolatos kutatások felölelik a modul referencia számláló modelleket, amelyek kilépéskor az aktivált rekordokat elérhetőség szempontjából ellenőrzik és „*well-stacking*" programok esetén hatékony módon regisztrálják, teljessé teszik és kiürítik az aktivált rekordok hulladékgyűjtését [62]. Ez a probléma, hatékony általános célú párhuzamos hulladékgyűjtés mellett [101] a jövő számítógép architektúráinál elveszítheti jelentőségét.

Deklaráció, közvetlen objektumlétrehozás ellen. Számos kutatási probléma merül fel az objektumok létrehozásával és deklarálásával kapcsolatban. Az azonosítóknak azt a deklarációját, amely a megnyilvánulás típusait megjelöli, nyelvi szinten világosan meg kell különböztetni az azonosítók olyan deklarációjától, amely az adott típusnak az objektumaira (eseteire) vonatkozik. Az azonosítóknak az újonnan létrehozott objektumokkal való összekapcsolása végrehajtható egy deklarációval, vagy egy értékadással, amely közvetlenül az explicit létrehozást követi, vagy — az olyan végrehajtható modulok esetében, mint az eljárások — egy hívással, amely egyesíti az objektum létrehozását a végrehajtással. (Az „eljárás-deklaráció" az input-output megnyilvánulás egy formáját definiálja, míg az „eljárás-hívás" létrehoz egy objektumot és azonnal indítja annak végrehajtását. Az invariáns eljárás kódon osztozkodhatnak az eljárás összes esetei, de az eljárás minden egyes esete számára szükség van az aktuális paraméter-objektumok és a lokális adatobjektumok egy-egy példányára.) Az objektum létrehozás és az összekapcsolás fogalmának az explicit különválasztása egy sokkal hajlékonyabb megnyilvánulást tesz lehetővé. Például az eljárások esetében az objektum létrehozása és a végrehajtás szétválasztása vezetett el a társrutin fogalmához. Azonban a deklarációval történő létrehozás világosabb programokat eredményezhet olyan esetekben, amikor az explicit létrehozás hajlékonyságára nincs szükség.

Az Algol 68 a „szigorúan vett" nyelvben megköveteli az objektumok explicit

létrehozását, de bevezet deklarációs létrehozást is a „kiterjesztett" nyelvbe mint rövidítést, olyan létrehozás „standard" használatára, amelyet értékadás követ. Az a gyakorlat, amely az olyan elkülönülő fogalmak számára, mint amilyen a létrehozás és az összekapcsolás, egymástól elválasztott mechanizmusokat hoz létre, miközben megenged rövidítéseket, amelyek az ortogonális fogalmak „standard" kompozíciói, jó nyelvtervezési gyakorlat.

Összekapcsolás fordítás közben erősen típusos nyelvekben. A deklarációkra vonatkozó egyik kutatási irány a speciális viselkedési formák fordítás közbeni összekapcsolásának erősségével foglalkozik. A szigorúan vett típus mind a programozónak, mind pedig a fordítóprogramnak értékes adatokat szolgáltat a program futás közben várható viselkedéséről, de bizonyos esetekben erősen korlátozhatja is a kifejező erősséget. Ennek a problémakörnek néhány kérdését a 3.1. fejezetben tárgyaljuk.

4.2.3. *A típus és a beburkolás kérdése*

A típus és a beburkolás problémaköre, az összekapcsoláshoz hasonlóan, az objektum modellezési kérdések egy speciális osztályaként is felfogható. A típusra és a beburkolásra vonatkozó kérdések felölelik a következőket:

a) Primitív típusok kiválasztása, típus kompozíciós és beburkolási módszerek.
b) Típusellenőrzés implementálása erősen típusos nyelvekben.
c) Beburkolt típusok számára nyelvi támogatást nyújtó módszerek.
d) Specifikációs technikák beburkolt típusok számára.
e) Beburkolás szekvenciális, párhuzamos és real-time programozásban.
f) Nagyon magas szintű nyelvek.
g) „Önmagukban elégséges" absztrakt környezetek.

Az *a)*, *b)*, *c)*, *d)* témákkal kapcsolatos kérdéseket az 1. és a 3. fejezetben tárgyaltuk. Az *e)*, *f)* és a *g)* problémakör rövid tárgyalását az alábbiakban adjuk meg:

Beburkolás szekvenciális, párhuzamos és real-time programozásban. A beburkolás egy nagyon hatékony módszer absztrakt viselkedési formák implementációtól független definiálására. A programozási nyelvi kutatások ezen a területen az absztrakt viselkedési formák partikuláris osztályainak az azonosításával és ilyen formák számára nyelvi támogatás nyújtásával foglalkoznak. WIRTH [64] az absztrakt viselkedési formák három osztályát azonosította: a szekvenciális programozásban fellépő absztrakciók (lásd 3.4. fejezet), a multiprogramozásban fellépő olyan absztrakciók, mint a monitorok (lásd 3.5. fejezet) és absztrakciók a real-time programozásban, amelyeket nemcsak mint a real-time programozás szinkronizációs feltételeit, hanem mint a real-time feltételeket is figyelembe kell venni.

Bizonyos implementációs kérdések közösek mind a háromféle absztrakciónál. Ilyenek: a kommunikáció kérdése az implementációs modul és annak felhasználói környezete között; az implementációs modulok létrehozásának és elindításának kérdése; a modulhatárok átlépésekor fellépő többletfeladatok minimalizálásának kérdése; az absztrakt modul viselkedések specifikálásának kérdése és annak kimutatása, hogy egy absztrakt viselkedési specifikáció konzisztens a komponensek viselkedési specifikációjával azon a modulon kívül, amelyben azt létrehozták. Az az út, ahogyan

ezekre a kérdésekre az absztrakciók speciális osztályaira vonatkozó válaszok megszületnek, betekintést nyújt az absztrakció természetébe. Az absztrakció folyamatának jobb megértése lehetővé teheti a nyelv tervezője számára azt, hogy az absztrakciót módszeresebben támogató nyelvi tulajdonságokat tervezzen és lehetővé teheti az alkalmazói programozó számára azt, hogy az absztrakciók olyan hierarchikus rendszerét fejlessze ki, amely az alkalmazások partikuláris osztályait rendszeresebb módon támogatja.

Nagyon magas szintű nyelvek. A nagyon magas szintű nyelvek (VHLL) a viselkedési formák nagyon magas szintjét támogatják — mintegy ellentétben a programozó által definiált típusokkal — primitív típusok segítségével. A VHLL által támogatott viselkedési absztrakciók azt specifikálják, hogy mit kell kiszámítani, függetlenül attól, hogyan kell azt kiszámítani és a fordítóprogramra bízzák a „mit" specifikáció „hogyan" implementációjának a kifejlesztését. A VHLL arról ismerhető fel, hogy támogatja a nem proceduláris megoldást (függetlenül az explicit vezérlési sorozattól), az adat tömörítést (függetlenül az adatstruktúra ábrázolásától) és az asszociatív címzést (annak a specifikációját, hogy mit kell elérni, függetlenül attól, hogyan történik az).

A hasznos, nagyon magas szintű absztrakciónak van egy felfogásában nagyon egyszerű „mit" specifikációja és egy ennél sokkal összetettebb „hogyan" implementációja. A VHLL kutatás általában az absztrakciónak azokra az osztályaira összpontosul, amelyeknek egyszerű „mit" és összetett „hogyan" specifikációjuk van. Az absztrakciónak a VHLL kutatásokkal kapcsolatos három ilyen osztályát azonosítjuk az alábbiakban:

1. A matematika egyszerűen specifikálható tulajdonságokkal foglalkozik, amelyeknek azonban összetett viselkedési hatásaik lehetnek. SETL [66] a VHLL egy olyan példája, amelyben vannak halmazok, mint alapvető adatösszességek és segédeszközök, amelyek természetes jelölést tesznek lehetővé a halmazelméleti okfejtések számára a matematikában.

2. A racionális adatbázis nyelvek [58, 67] annak a specifikálását tartják szem előtt, hogy mit kell lekérdezni az adatbázisból, függetlenül attól, hogy a lekérdezendő objektumok hogyan vannak ábrázolva az adatbázisban és felfoghatók olyan VHLL nyelvekként, amelyek támogatják az adatbázisokra vonatkozó viselkedések absztrakt elérését.

3. A mesterséges intelligencia nyelvek [3, 68] magukba foglalhatnak nagyon magas szintű nyelvi mechanizmusokat az adatösszességek, az implicit vezérlési sorozat, az asszociatív címzés, az alakfelismerés, a procedurális megismerés és a deduktív okfejtés specifikálása számára. Azok a mesterséges intelligencia problémák, amelyek játékoknál és tételbizonyításnál fellépnek, gyakran komplex implementációval rendelkező szabályok („mit" specifikációk) egy egyszerű halmazával jellemezhetők. Ezek a problémák gyakran egy állapottér alapján fogalmazhatók meg, amelyben van egy kezdeti állapot, a célállapotok egy halmaza, egy állapottranszformációs szabály és az a követelmény, hogy a kezdeti állapotból egy célállapotba vezető ösvényt találjunk. A sakkjáték pontosan illeszkedik ehhez a példához és szemlélteti azt a nagy potenciális egyenlőtlenséget, amely a specifikációs szabályok egy halmaza és a szabályok egy halmaza által specifikált feladat végrehajtására vonatkozó stratégia között fennáll.

A VHLL kutatás egyaránt foglalkozik olyan absztrakciók fajtáinak a meghatározására szolgáló specifikációk problémáival, amelyeket hasznos lehet VHLL nyelvekkel támogatni és olyan implementációs és optimalizációs problémákkal, amelyek a VHLL nyelvek fordítóprogramjainak, értelmezőinek a kifejlesztésénél fellépnek. A VHLL nyelv implementációját úgy tekinthetjük, mint egy olyan programszintézis problémát, amelyet a nyelv implementációját végző személynek kell megoldania.

Önmagukban elégséges absztrakt környezetek. Úgy tűnik, hogy nagyon nehéz kifejleszteni olyan általános célú absztrakt formalizmusokat, amelyek lehetővé teszik a számításba vett alkalmazási területtel kapcsolatos összes okfejtés végrehajtását absztrakt szinten, az absztrakció alacsonyabb szintjeire való leereszkedés nélkül. Ez a nehézség vezetett bizonyos alkalmazásoknál az assembly nyelvek folyamatos népszerűségéhez és annak a szükségességéhez, hogy az erősen típusos nyelvekben legyenek a típusnál „kiskapuk". Egy absztrakt formalizmus nem zárt voltának problémája nemcsak a teljes nyelvek szintjén lép fel, hanem az egyes típusok szintjén is, ahol általában a túlcsordulást és más hibafeltételeket definiáljuk speciális esetekként és nem pedig az absztrakt típusviselkedések integrális részeként. A viselkedési absztrakciók egy rendszerének az önmagában való elégségessége nemcsak azt követelheti meg, hogy megengedett műveletek mellett a típusokat beburkoljuk, hanem azt is, hogy gazdagítani tudjuk a típusok megnyilvánulásait olyan új típusműveletek hozzáadásával, amelyeket a típusok halmazának definiálása idején még nem ismertünk fel. Például, a vezérlési absztrakciók [44, 45] olyan mechanizmusoknak tekinthetők, amelyek gazdagítják (kiterjesztik) egy absztrakt adattípushoz tartozó műveletek halmazát a típus elemeinek sorozatára vonatkozó műveletek hozzáadásával.

4.2.4. *Vezérléssel kapcsolatos kérdések*

Egy vezérlési struktúra az akciókra vonatkozó viselkedési formák egy absztrakt sorozatát határozza meg. Ez hasonlít a típus esetéhez, amely az objektumok osztályának egy absztrakt viselkedési formáját határozza meg. Az alábbiakban megvizsgálandó kutatási területek a következők:

a) Vezérlési struktúrák egy állandó elérésű környezetben.
b) A modulhatárokon keresztül történő átvitel.
c) Az elfogadás kezelése.
d) Nem determinisztikus vezérlési struktúrák.

Állandó elérésű környezetek. Egy állandó elérésű környezethez tartozó vezérlési struktúrákat a 3.2. fejezetben tárgyaltuk és tárgyalásuk [36]-ban található meg, amely egyaránt tekintetbe veszi a nyelv tervezésének kérdéseit és a vezérlési struktúrákra vonatkozó algoritmikus hatékonyság kérdéseit.

A modulhatárok átlépése. A határok átlépésére vonatkozó vezérlési mechanizmusok az elérhető objektumok halmazában változásokat eredményezhetnek és támogatják az absztrakció szintjeinek a váltását. Ezek a vezérlési mechanizmusok eljáráshívásokat és visszatéréseket, társrutin-hívásokat és újrakezdési parancsokat, továbbá az elfogadást kezelő mechanizmusokat ölelik fel.

Az eljáráshívásoknak létre kell hozniuk egy új környezetet, és gondoskodniuk kell a visszatérésről, amely a hívott környezetet megszünteti. A társrutinoknak ki-

lépéskor szükségük van a környezet és az aktuális adatok megőrzésére, hogy azokat folytatni tudják. Az absztrakt adatmodulokban levő eljárások hívásai megkövetelik a modulkörnyezetbe való belépést, amelyet az eljáráskörnyezetbe való belépés követ. Az eljárásból való kilépéskor megengedett az eljáráskörnyezet megszüntetése, a modulkörnyezetet viszont a modulból történő kilépéskor meg kell őrizni. A Simula-szerű beburkolt class struktúrák, amilyenek a konkurrens Pascalban [49] is előfordulnak, egy modulon belül támogatást nyújtanak az eljárásoknak, társrutinoknak és az adatabsztrakciónak. Ebben a problémakörben a nyelvtervezési kérdések egyik problémája, hogy vajon az eljárás-, a társrutin- és az adatabsztrakciót egymástól elkülönített beburkoló mechanizmussal kell-e támogatni, vagy egy nyelvi mechanizmussal.

A kivételek kezelése. A kivételek kezelése a hívott rutinból jövő szokásostól eltérő vagy hibás visszatérések kezelését jelenti. A kivételeket nagyon fontos explicit módon specifikálni, mivel ez teszi lehetővé a szokásostól „eltérő" és hibás feltételek esetén is a modul viselkedésének ugyanazon az absztrakciós szinten való kezelését, mint normális esetben. Az absztrakt modul specifikációjának részét képező kivételes viselkedés konzisztens kezeléséhez szükséges, hogy a hívott modul kivételes feltételeit maga a hívó modul kezelje. Ez jobb megoldás, mint amikor ezt egy ún. ON feltétel teszi meg, amelyet a környezet egy tetszőleges külső rétegében deklarálunk (például a PL/1-ben). Az eljárások céljára szolgáló kivételeket kezelő mechanizmus tervezésekor a *folytatásos modell* és a *terminációs modell* közül kell választani. A folytatásos modell megengedi, hogy az eljárás jelzést adjon a folytatásra a kivétel kezelésekor. A terminációs modell szokásostól eltérő kilépés esetén az eljárást éppen úgy befejezi, mint a normál kilépés esetén. Vannak, akik [70] amellett érvelnek, hogy a folytatásos modell nyelvi és felfogásbeli bonyolultságának többletkülönbsége talán nem is éri meg. A kivételek kezelésének további tárgyalása [69, 70, 71]-ben található.

Nem determinisztikus vezérlési struktúrák. A nem determinisztikus vezérlési struktúrák két különböző alkalmazási összefüggésben merülnek fel:

1. Olyan problémák esetén, amelyeknél egy utat kell megkeresni egy kezdő állapottól a célállapotig egy olyan állapottérben, amelyben minden állapotot több másik állapot követhet. Az állapottér felderítése végrehajtható olyan primitívák [61] segítségével, amelyek megengedik bármely adott állapot esetén az azt követő összes állapot párhuzamos felderítését.
2. A nem determinisztikus tulajdonságok modellezésének vezérlése a számítógéprendszerekben is felmerül. Az ilyen nem determinizmus a *Dijkstra-féle* ún. *guarded kommandokkal* modellezhető, amelyeknek formája a következő:

$$p_i : a_i,$$

ahol p_i predikátum, amely „őrzi" az a_i tevékenységet, olyan értelemben, hogy az csak akkor hajtható végre, ha p_i igaz. DIJKSTRA bevezette a következő formájú általánosított if utasítást:

$$\underline{\text{if}}\ p_1 : a_1 \square \ldots \square p_n : a_n\ \underline{\text{fi}},$$

amely nem determinisztikus módon hajt végre egy olyan a_i tevékenységet, amelyhez igaz értékkel rendelkező p_i tartozik és hibás utasítást eredményez, ha nincs ilyen p_i.

Dijkstra egy általánosított while utasítást is bevezetett a következő formában:

$$\underline{\text{do}}\ p_1 : a_1 \square \ldots \square p_n : a_n\ \underline{\text{od}},$$

amely egy iteráció, és egymás után olyan a_i utasítások végrehajtását jelenti, amelyekhez igaz értékkel rendelkező p_i-k tartoznak és ha nincs ilyen p_i, akkor az üres utasítást eredményezi. Ezen a területen folyó kutatások egyrészt programozás-módszertani kutatások, másrészt programverifikációs kutatások. A módszertani kutatások célja az ilyen konstrukciók hatékony használatának az elérése. A verifikációs kutatások olyan programok helyességének a bizonyításával foglalkoznak, amelyek ilyen konstrukciókat használnak [60]. Azokat a nem determinisztikus programokat, amelyeknek a számítási hatása független a végrehajtás alatti nem determinisztikus választástól, determinisztikus programoknak nevezzük. Folynak olyan kutatások, amelyek annak a kimutatásával foglalkoznak, hogy a nem determinisztikus programok bizonyos osztályai determinisztikusak.

Mind a kétféle nem determinisztikus vezérlési struktúra a problémaspecifikáció jelentős egyszerűsítését eredményezheti. Ezért ezeket olyan nagyon magas nyelvi mechanizmusoknak tekinthetjük, amelyek megengedik a felhasználónak, hogy közömbös helyzetekben a döntés meghozatalának a terhét a programozási rendszerre hárítsa át. Közömbös helyzetekben a döntés meghozatal elkerülésének a képessége lényegében az egyik leghatékonyabb mechanizmus, amely komplex feladatok kezelhető részekre való redukálására szolgál és számos esetben a feladat komplexitását exponenciálisról lineáris komplexitású részekre redukálhatja. A nem determinisztikus vezérlési struktúrákkal, a nem determinisztikus problémaspecifikációval és az absztrakciók beburkolására szolgáló mechanizmusokkal (nem determinisztikus esetről van szó) kapcsolatos kutatások eszközt szolgáltathatnak a mesterséges intelligenciában ma még igen nehéz problémák specifikációjának az egyszerűsítésére.

Dijkstra [60] rámutatott, hogy a nem determinisztikus programokkal kapcsolatos következtetések sokkal szövevényesebbek, mint a determinisztikus programokkal kapcsolatos következtetések. Ez azért van, mert egy adott kezdeti állapot a számításoknak (számítási fáknak) egy olyan halmazát idézi elő, amelyben helyesen befejeződő, helytelen eredménnyel befejeződő és be nem fejeződő számítások vannak. A nem determinisztikus programok logikájával kapcsolatos elméleti kutatások [73], nem determinisztikus számításokkal kapcsolatos következtetések számára fejlesztenek ki egy olyan sémát, amely különbséget tesz végtelen számítások, valamint hibák között és analizálják az ilyen sémában megadott *Dijkstra-féle „leggyengébb előfeltétel"* logikát. (Egy R megkívánt utófeltétellel adott S utasítás $wp(S, R)$ leggyengébb előfeltétele azon előfeltételeknek a halmaza, amelyek garantálják, hogy az S utasítás végrehajtása egy R által meghatározott állapotban fog befejeződni.)

4.3. *Elméleti kutatások*

4.3.1. *Programozási nyelvi modellek*

Az elméleti modellek kifejlesztéséhez a programozási nyelvek területe termékeny talajnak bizonyult. Bár a modellek hasznossága és mélysége vitatható, lehet érvelni azzal is, hogy hiányzanak a mély eredmények, mindazáltal a programozási nyelvek által létrehozott modellek választéka vetekszik a fizikai és matematikai tudományok modelljeinek választékával. A modellek következő osztályait különböztethetjük meg.

1. Formális nyelvek modelljei és automata elméleti modellek [76], amelyek elsősorban szintaxisokkal foglalkoznak és a fordítóprogramok kidolgozásánál találtak gyakorlati alkalmazásra.

2. Szemantikai modellek, amelyek egységes módszert nyújtanak egy programozási nyelv bármely P programja esetén a program $M(P)$ jelentésének a meghatározására. Nincs általános megállapodás arra nézve, hogy egy programozási nyelv jelentésén mit kell érteni, ezért a szemantikai modelleknek számos különböző osztálya létezik, amelyeknek mindegyike a jelentés egy-egy fogalmához és a megnyilvánulási absztrakció egy-egy módszerével társul. A megjelölő (denotációs) modellek [116] állnak legközelebb a jelentés tiszta „plátói" fogalmának megragadásához. A modellek más fontos részosztályai a következők:

3. Az axiomatikus és algebrai modelleket a programok specifikálásánál, helyességbizonyításánál és transzformálásánál használjuk. Létezik a szemantikai modelleknek egy olyan részosztálya, amely formális módszerrel rendelkezik összetett struktúrák jelentésének komponenseik jelentése alapján történő kifejezésére.

4. Nyelvi definíciós modellek [77, 109] a teljes nyelvek definíciójával foglalkoznak, olyan formában, hogy az a felhasználók, tervezők és az implementációt végzők számára hasznos. Ezek a szemantikai modellek általában műveleti operációs modellek abban az értelemben, hogy ezek a programok jelentését egy absztrakt számítógépen történő kiszámítás formájában definiálják.

5. Blokkdiagram modellek és programsémák [75] a program végrehajtásában a vezérlés folyamatának a tanulmányozására helyezik a hangsúlyt, függetlenül a lényeges utasítások által specifikált számításoktól.

6. A lambda kalkuluson alapuló modellek a paraméter átadás, behelyettesítés és a programozási nyelvek absztrakt mechanizmusának lényegébe nyújtanak betekintést.

A modellek mindegyik osztálya egy-egy jellemző absztrakciós mechanizmussal társul és lehetővé teszi számunkra a programozási nyelv egy részhalmazában való elmélyedést a többiek ismerete nélkül. A kutatási irányok tárgyalásánál mi az axiomatikus és az algebrai modelleket fogjuk hangsúlyozni, mivel — úgy tűnik — ezeknek van a legközvetlenebb hatása a nyelv tervezésére és a programozási módszertanra.

4.3.2. *Objektumokra és tevékenységekre vonatkozó specifikációs sémák*

A specifikáció annak a pontos és teljes jellemzésével foglalkozik, amit ki kell számítani. A specifikáció természetét részben a specifikálandó megjelenési forma belső természete határozza meg, ezért a tevékenységekre vonatkozó specifikációk különböznek az objektumokra vonatkozó specifikációktól. A következő témákkal kívánunk foglalkozni:

a) Nyelvi konstrukciókra vonatkozó specifikációs sémák.
b) Objektumok specifikálása algebrák segítségével.
c) Specifikációk helyessége és teljessége.
d) Absztrakt specifikációk műveleti változatai.

Specifikációs sémák. Az a tevékenység, amelyet az S utasítás meghatároz — *Hoare-féle jelölés* szerint — a $\{P\}\ S\ \{Q\}$ formulával specifikálható, ahol P az

S utasítás végrehajtása előtti számítási állapotra vonatkozó előfeltétel, Q pedig az S végrehajtása utáni állapot „utófeltétele". Például az

$$\{x = 9\}x := x+1 \ \{x = 10\}$$

formula azt specifikálja, hogyha a számítási állapot x komponense 9 volt az $x:=x+1$ utasítás végrehajtása előtt, akkor az az utasítás végrehajtása után 10 lesz. A számítási állapotnak azok a komponensei, amelyeket a P és Q nem érinti, változatlannak tekintendők.

Fontos, hogy megkülönböztessünk egymástól egyes programokra vonatkozó specifikációkat, amelyekkel [78] foglalkozik és nyelvi konstrukciókra vonatkozó specifikációs sémákat, amelyek a konstrukciók eseteinek specifikálására vonatkozó szabályokat határoznak meg. Például az értékadás szemantikáját definiálhatjuk a

$$\{P_E^x\}x := E\{P\}$$

specifikációs sémával, amely azt specifikálja, hogy az $x:=E$ végrehajtása utáni P utófeltétel maga után vonja a P_E^x előfeltételt, amelyet akkor kapunk, ha P-be az x helyébe E-t helyettesítünk be. Ennek a specifikációs sémának a speciális esete $\{x+1=10\}x:=x+1\{x=10\}$, ahol $E=x+1$ és P az $x=10$ feltételnek felel meg.

A „while B do S" konstrukciót a

$$\text{„}\{P \wedge B\}S\{P\}\text{-ből következik } \{P\} \text{ while } B \text{ do } S\{P \wedge \neg B\}\text{"}$$

specifikációs sémával specifikálhatjuk, amely a while utasítás viselkedését definiálja a P utasítás formájában, amelyet az S utasítás végrehajtása változatlanul hagy. Ez a specifikáció annak az ötletnek a felhasználása, hogy az S utasítás viselkedése nemcsak azzal jellemezhető, hogy mit változtat meg, hanem azzal is, hogy mit hagy változatlanul. A „while" viselkedését egy olyan számítási sorozat jellemzi, amelyet a P invarianciája szorít korlátok közé és amelynek célja a $\neg B$ feltétel elérése. Az invariancia az S utasítás egy kiegészítő viselkedésére példa, amelyet minden egyes while utasításra egyedileg kell felismerni. Specifikációs sémákból specifikációk származtatása általában bizonyos „kreativitást" követel meg. A while utasításokra vonatkozó specifikációs séma ad némi támpontot ahhoz, hogy mit kell megvizsgálni a while ciklus jelentésének a felismeréséhez egy adott S utasítás esetén.

Ezen a területen a kutatás a programozási nyelvi konstrukciók számára olyan specifikációs sémák kifejlesztésére törekszik, amelyek a programozók számára szolgáltatnak vezérfonalat programjaik és programkomponenseik specifikációjának a megkonstruálásában. Különösen hasznosak azok a specifikációs sémák, amelyek a programozó számára olyan hasznos segédkonstrukciók bevezetéséhez nyújtanak útmutatást, mint amilyenek az invarianciák. A specifikus nyelvi konstrukciókkal kapcsolatos kutatásokat jól szemléltetik az eljárásokkal [81, 82], az utasításokkal [83], valamint a vezérlésátadásokkal és a függvényekkel [84] kapcsolatos kutatások példái. A Pascal [85] és az Euclid [86] formális specifikációi mutatják, hogyan egyesíthetők az egyedi nyelvi tulajdonságok egy teljes programozási nyelv specifikációjában.

Objektumok specifikálása algebrák segítségével. Ezt a témát a 3.5. fejezetben tárgyaltuk és ezzel foglalkozik a [88] munka is. Ezen a területen folyó jelenlegi kutatások felölelik a hiba absztrakcióit [87], amely lehetővé teszi a hibák, a kivételek és a műveletek nem zárt voltának az absztrakt kezelését. Még általánosabban, kutatásra van szükség egy absztrakt környezet specifikációs és nyelvi támogatására, amely meg-

engedi, hogy az absztrakt objektumok összes megnyilvánulását absztrakt módon, implementációtól függetlenül kezeljük. Olyan kiterjesztési mechanizmusokat kell beépíteni a nyelvekbe az absztrakció támogatására, amelyek megengedik új műveletek hozzáadását az absztrakt objektumokhoz.

Az absztrakt algebrák területén folyó jelenlegi kutatások célja az adatabsztrakciók számára matematikai bázis létrehozása. Egy ígéretes ötlet [88] az összetett specifikációk hierarchikus konstrukciójának a gondolata, amely párhuzamosan hajtható végre az összetett modulok moduláris összetevőkből történő hierarchikus konstrukciójával. A [88] munka olyan algebrákkal foglalkozik, amelyekben a műveletek közötti összefüggések, mint tételek szerepelnek, a tételekben műveleteket definiál, például a tétel kibővítése új adattartományokkal, újfajta adatokkal, műveletekkel és egyenlőségi összefüggésekkel, bevezet továbbá eljárásokat, amelyek a tételekkel végezve műveleteket, új tételeket hoznak létre. A Clear programozási nyelv [88], amely a tételekre vonatkozó fenti műveleteket támogatja, maga is rendelkezik tételekkel, adattípus specifikációkkal mint adatobjektumokkal és az adatobjektumain fontos műveletek végrehajtására képes.

Specifikációk helyessége és teljessége. A helyesség és a teljesség az absztrakt specifikáció és eseteinek szóba jöhető ekvivalens osztálya között fennálló összefüggéssel jellemezhető, lásd az 1.3. fejezetet. (A helyesség és a teljesség fogalma a logikából származik, de közvetlen szerepet játszik a formális specifikációban. A helyesség és a teljesség a specifikáció adekváltságával foglalkozik egy elérendő interpretáció megvalósításában és megköveteli az interpretáció független szemantikai jellemzését, mint ahogyan a predikátum kalkulus sem nélkülözheti az igazság fogalmának egy független jellemzését.) A specifikációt akkor mondjuk helyesnek, ha az általa meghatározott eseteknek az ekvivalens osztálya az elérendő ekvivalencia osztálynak egy részhalmaza; és teljesnek akkor mondjuk, ha az általa meghatározott esetek ekvivalens osztálya az összes elérendő eseteket felöleli. Eszerint az asztalszerűség specifikációja akkor helyes, ha minden objektum, amely a specifikációt kielégíti asztal, és akkor teljes, ha az összes asztal a specifikáció egy esete. Az a specifikáció, amely helyes is, és teljes is, rendelkezik azzal a tulajdonsággal, hogy az esetek elérendő ekcivalencia osztálya pontosan egybeesik a specifikált ekvivalencia osztállyal.

A jelenlegi kutatások az absztrakciók helyes és teljes specifikációjának a kifejezése szempontjából a specifikációs nyelvek által nyújtott lehetőségek korlátaival foglalkoznak. Sajnos, a programokra vonatkozó megállási problémának az eldönthetetlensége maga után vonja, hogy az olyan specifikációs nyelv, mint a predikátum kalkulus, alkalmatlan arra, hogy egy programozási nyelv minden programja számára konstruktív, helyes és teljes input-output specifikációt szolgáltasson. Kimutatható [89], hogyha egy előrejelzést bevezetünk, amely a megállással kapcsolatos kérdésekre választ ad, akkor olyan egyszerű programozási nyelvekre, amelyek csak értékadást, if-then-else, vagy do-while konstrukciót tartalmaznak, helyes és teljes specifikációs nyelvek fejleszthetők ki. Bár a legújabb kutatások [90, 91] kimutatták, hogy bizonyos nyelvi konstrukcióknak, mint a név szerint hívott eljárásparamétereknek még az ilyen előrejelzéssel ellátott specifikációs nyelvekben sem lehet helyes és teljes specifikációjuk. Ez a kutatási eredmény jelentős, mivel lehetővé teszi, hogy külön válasszuk azokat a specifikációs problémákat, amelyek a matematikai és a programozási nyelvek között fennálló egyenlőtlenségekből származnak, azoktól a problémáktól, amelyek abból a nehézségből adódnak, hogy magas szintű nyelvi konstrukciókat alacsony szintű számítási primitívákkal fejezünk ki.

4.3.3. *Verifikáció*

A programverifikációs technikák alapját azok a formális rendszerek képezik, amelyeknek axiómái (axióma sémái) a nyelvi konstrukciók specifikációs sémái és amelyeknek a tételei speciális programok input-output specifikációi. Például, az értékadás

$$\{P_E^x\}x := E\{P\}$$

axiómasémája lehetővé teszi számunkra az $x := x+1$ program esetében a

$$\{x = 9\}x := x+1\{x = 10\}$$

tétel levezetését.

A specifikációs sémákat olyan következtetési szabályokként is fel lehet fogni, amelyeknek segítségével a komponens struktúrákra vonatkozó tételekből összetett struktúrákra vonatkozó tételek vezethetők le. Például a while utasításra vonatkozó $\{P \wedge B\}S\{P\}$-ből következik $\{P\}$ while B do S $\{P \wedge \neg B\}$ specifikációs séma lehetővé teszi az S utasításra vonatkozó tétel alapján a while B do S utasításra vonatkozó tétel bebizonyítását. A következtetési szabály egy másik példája $\{P\}\, S_1\, \{Q\}$ és $\{Q\}\, S_2\, \{R\}$-ből következik $\{P\}S_1; S_2\{R\}$ specifikációs séma, amelynek segítségével a komponensekre vonatkozó tételekből a két utasítás kompozíciójára vonatkozó tétel vezethető le. A következtetési szabályokat néha bizonyítási szabályoknak is szokták nevezni, mivel ezeknek segítségével a komponensekre vonatkozó tételek alapján összetett programszerekezetekre vonatkozó tételek bizonyíthatók be.

A programverifikáció kutatások egyrészt a programhelyesség kritériumának specifikációs kérdéseivel, másrészt a tételbizonyítás implementációs problémáival foglalkoznak. A specifikációs problémáknak az a gyökere, hogy egyrészt a számítási feladatok pontos és teljes leírására szolgáló képességük nagyon korlátos, másrészt pedig az alkalmazói programokban fellépő lényegesen komplex számítási feladatok specifikációi kezelhetetlenül bonyolultak is lehetnek. A tételbozonyítási problémák közé a következők sorolhatók:

— a tartományfüggetlen kombinatorikus problémák, amelyek bármely tételbizonyítási alkalmazásnál felléphetnek;

— olyan tartománytól függő speciális problémák, mint a bizonyítás során generált algebrai kifejezések egyszerűsítésének kérdése, invariánsok előállítása és más segédabsztrakciók létrehozása azzal a céllal, hogy a konzisztencia bizonyítás kombinatorikus kezelhetetlenségét javítani tudjuk [92].

Az automatikus bizonyítási módszereket a [93] tanulmány tárgyalja, a programverifikáció kutatási eredményeinek részleteit pedig a [95] cikk ismerteti.

A programok a számítási viselkedés dinamikus jellemzői, a specifikációk pedig a számítási állapotokban megadott megnyilvánulás statikus jellemzői. A bizonyítás elősegítése céljából hasznos megengedni a programozónak, hogy a program közbeeső pontjaiban a számítási állapotra vonatkozó állításokat helyezzen el. Az ilyen állításokkal ellátott program leírásához a dinamikus programozási nyelvet egy statikus állítások leírására alkalmas nyelvvel kell kibővíteni. Még tovább is mehetünk, olyan programozási nyelvre is gondoljunk, amely nemcsak az állítások elhelyezését engedi meg, hanem be is bizonyítja, hogy a program a programban elhelyezett specifikáció szerint helyes [96]. Ez a szemlélet a specifikációval, kiegészítő állításokkal és bizonyításokkal kiegészített programot egy olyan egyetlen formális objektumnak

tekinti, amely két kiegészítő programspecifikációt és azok ekvivalenciájának a bizonyítását tartalmazza. Bevezetve egy ilyen formális objektum fogalmát, a verifikációt egy olyan objektum szintetizálásának a folyamataként foghatjuk fel, amelyet egy teljes dinamikus és egy nem teljes statikus specifikációval ellátott parciálisan specifikált objektumból állítunk elő, a programszintézis pedig olyan objektum szintetizálásának a folyamata, amelynél a dinamikus specifikáció nem teljes.

A $\{P\}\ S\ \{Q\}$ specifikáció az S szemantikáját a P és Q predikátumok kapcsolatával definiálja. (Az S utasítás a predikátumtranszformátor egy formális állapotával rendelkezik, amelyben az S szemantikája az utasítás végrehajtása által meghatározott predikátumtranszformációval teljesen jellemezhető.) Az ilyen specifikáció kétirányúan is felhasználható: P ismeretében meghatározható Q és adott Q utófeltétel esetén meghatározható azoknak a P előfeltételeknek a halmaza, amelyek a Q előállítására képesek. Az előbbi eset csak az S parciális helyességének a kimutatására használható fel. Ilyenkor S befejeződését feltételezzük és a befejeződés kimutatásához ezt a következtetést ki kell egészíteni egy ettől független módszerrel [100]. A visszafelé való következtetést alkalmazó verifikációs módszerek közé tartozik DIJKSTRA a leggyengébb előfeltételre épülő megoldása [60], amely axiómákat és következtetési szabályokat ad meg az összes olyan előfeltételek $wp(S, Q)$ halmazára, amelyek az S utasítás befejeződését eredményezhetik abban az állapotban, amelyben Q fennáll. A visszafelé való következtetés logikailag vonzó, mivel az nem kívánja meg a befejeződés elkülönített kezelését és azonos logikai vázlat alapján dolgozik a determinisztikus és a nem determinisztikus programokkal. A programverifikációra szolgáló számítógépes rendszerek azonban elsősorban a másik megoldást, az előre történő következtetést használják.

A programverifikációt a matematikai logikai rendszerek tanulmányozása alapozta meg. A matematikai logika javasolt rendszerei között szerepel a strukturális indukció módszere [87], amelynél a program adott pontjában olyan állítást helyezünk el, amely valamikor igaz lesz. Ez a módszer a program állapotára vonatkozó erősebb állítások elhelyezését engedi meg, mint más módszerek. CONSTABLE viszont a verifikációhoz a konstruktív logikát használja fel bázisként [96]. Pratt [98] a dinamikus logikát vezette be az

„S végrehajtása utáni állapot"

(jelölése $[S]$) fogalmának, mint logikai primitívának a bevezetésével. MILNER [99] a kiszámítható függvényekre (LCF) fejlesztett ki egy logikát, amely a kombinatorikus logikán és a lambdakalkuluson alapszik. A logikának a verifikációban és a szemantikában játszott szerepére vonatkozó áttekintés [112]-ben található meg. A párhuzamos programok verifikációs módszereibe való bevezető munka a [101], amely a hulladékgyűjtés egyik módszerét használja fel példaként.

4.3.4. *Programtranszformáció és programszintézis*

A *programtranszformáció* a program szövegének olyan szintaktikus átalakítását jelenti, amelynek során a szemantikai ekvivalencia fennmarad. Az itt folyó kutatások nagyon fontos betekintést adnak a program szerkezetébe, amely nemcsak az implementáció hatékonyságának a bizonyításában hasznos, hanem a nyelv tervezésében és a programozási módszertanban is.

A programtranszformációkat specifikálhatjuk egyenletekkel, mint például $x+y=y+x$, amely mindkét irányban megengedi a helyettesítést, vagy produkciókkal, mint (if true then S else T)$\rightarrow S$, amely csak egy irányú helyettesítést foglal magába. A produkciók célirányú transzformációk definiálására is használhatók [102]. Egy nagyon fontos alkalmazása a célirányú transzformációnak a következő: a felfogásukban egyszerű, de talán kevésbé hatékony programoknak az átalakítása bonyolultabb, de hatékony programokká. A jellemző példák között megemlíthetjük a fordítóprogram optimalizálását adatfolyam analízis segítségével [103, 107] és a rekurzió kiküszöbölését [104].

Kis lokális transzformációk sorozatát összeláncolhatjuk abból a célból, hogy globális tulajdonságú eredő transzformációkat érjünk el. Ezt a problémát és ennek egy érdekes alkalmazását tárgyalja a [105] munka. A bemutatott alkalmazás az általános mátrixszorzást végző program optimalizálása olyan speciális tulajdonságú mátrixok esetére, mint például a háromszög mátrixok. Ez a példa mutatja, hogy a programtranszformáció a fordítóprogramoknál nemcsak mint egy problémától független technika, hanem mint egy problémától függő technika is hasznos, amely lehetővé teszi különleges algoritmusok — mint amilyen a mátrixszorzás — „ráhangolását" speciális esetekre.

A programszintézis az absztrakt problémaspecifikációnak olyan programmá történő átalakításával foglalkozik, amely a specifikációt megvalósítja. Ezt úgy is felfoghatjuk, mint a célirányú programtranszformáció egy speciális esetét egy olyan nyelv esetében, amely a specifikációs nyelvnek és a programozási nyelvnek az egyesítése. A problémát eredetileg az egyesített nyelvnek a specifikációra szolgáló részhalmazával specifikáljuk. A célunk pedig az, hogy ezt a specifikációt közbeeső ábrázolásokon (amelyek lehetnek részspecifikációk és részprogramok) keresztül átalakítsuk teljesen a meglehetősen gazdag nyelvnek a programozási nyelvi részhalmazára. A [108] munka jó példa az ilyen indítékú kutatásra.

Nagyon magas szintű nyelvi konstrukciók a programtranszformációs szabályokon keresztül alacsonyabb szintű megvalósításokkal kapcsolhatók össze. A makrók és általában bármely fordítóprogram is, felfogható úgy, mint egy programtranszformációs rendszer, amely a forrásnyelvű programszövegen végez műveleteket azzal a céllal, hogy létrehozza a célnyelvű programszöveget. Olyan programtranszformációs rendszert, amely tetszőleges műveletsorozatot valósít meg, viszonylag könnyű konstruálni.

Az ilyen rendszer célja az, hogy feltárja a szemantikailag ekvivalens programok ekvivalens osztályainak a szerkezetét és a „navigáció" célirányú módszereit. Az a tény, hogy a programekvivalencia parciálisan nem dönthető el, mutatja, hogy a navigáció problémája egy tetszőleges A elemtől egy vele ekvivalens B elemig, vagy egy kívánt tulajdonságú X elemnek a megkeresése, soha sem oldható meg maradéktalanul. Mindamellett azonban az olyan transzformációk, amelyek speciális navigációs segédeszközöket nyújtanak, speciális helyzetekben, ennek a területnek nagyon értékes kutatási eredményei.

4.3.5. *Szemantika*

A programozási nyelv szemantikája az adott nyelv programjainak a specifikációját adja, de nem szükségképpen olyan formában teszi ezt, amely konstruktív vagy a program helyességének a bizonyítására alkalmas. A szemantikával kapcsolatos kutatások felölelik a verifikációs célt szolgáló specifikációs modellek tanulmányozását, de foglalkoznak a program jelentésének a kérdéseivel egy szélesebb perspektíva szempontjából is. A szemantika modellt egy $M=(E, D, O)$ hármassal definiálhatjuk, ahol E a kifejezések (programok) szintaktikus tartománya, D az értelmezések (jelentések) szematikus tartománya és O egy szemantikus leképező függvény, amely a kifejezéseket azok értelmezéseire képezi le. Az értelmezések D tartománya a programok elfogadható jelentéssel bíró osztályát határozza meg és alapul szolgál a szemantikus modellek osztályozásához. Az O leképezést rendszerint szabályokkal adjuk meg: olyan szabályokkal, amelyek egyszerű szintaktikus objektumokat egyszerű szemantikus objektumokra képeznek le és szemantikus kompozíciós szabályokkal (következtetési szabályokkal), amelyek a komponensek leképezései alapján definiálják a szintaktikailag összetett objektumokra a leképezést.

Megkülönböztetünk

— műveleti modelleket,
— megjelölő (denotációs) modelleket,
— axiomatikus modelleket, és
— algebrai modelleket.

A műveleti modell az implementáció egy absztrakt modelljének segítségével specifikálja a jelentést. A megjelölő modellekben parciális rekurzív függvények alkotják az értelmezések D tartományát. Az axiomatikus modellben predikátum transzformációk formájában adjuk meg az S tevékenység jelentését. Végül az algebrai modell az adatabsztrakciók jelentését algebrákkal specifikálja. Az operációs modellt McCarthy vezette be a Lisp [22] definiálására és ez vezetett a *Bécsi Definíciós Nyelvhez* VDL-hez [109], valamint annak utódaihoz, ilyen például [110]. A megjelölő modelleket Scott és Strachey vezette be [111] és ezeket számos újkeletű könyv [113, 114, 115], valamint cikk [116, 117] tárgyalja. Az axiomatikus modelleket először Floyd [79] és Hoare [80] munkáiban találjuk meg. Az algebrai modellek még csak csecsemőkorukat élik. Tárgyalásukat megtalálhatjuk a [88] munkában és ezek szolgáltatják az adatabsztrakcióval kapcsolatos munka alapját is [44, 45].

A programozási nyelv szemantikája nem egy abszolút (plátói) fogalom. A jelentésnek több egyformán érvényes fogalma létezik, ezeket azok a különböző szövegösszefüggések határozzák meg, amelyekben a jelentés fogalmát használni kell. A műveleti modellek a jelentésnek egy nagyon konkrét fogalmát adják és ezek közelíthetik meg leginkább a jelentésnek azt a fogalmát, amely a nyelv tervezőinek a képzeletében létezik. De ezek, más műveleti specifikációkhoz hasonlóan eltűrik azt, hogy lényegtelen részleteket és szükségtelen komplexitást tartalmaznak. A megjelölő modellek jutnak el legközelebb a programok tiszta (plátói) jelentésének a specifikálásához. Amint azt De Bakker kimutatja [117] a megjelölő szemantika, a műveleti szemantikához hasonlóan, modellt használ a jelentések specifikálására, de a műveleti primitívákkal ellentétben, matematikai primitívákkal dolgozik. A megjelölő szemantika bevezeti

— az állapot fogalmát, mint a helyekről az értékekre való leképezést;
— a környezetnek a fogalmát, mint a változókról a helyekre való leképezést;
— a parancsok fogalmát, mint az állapotokon értelmezett leképezést, és
— a kifejezések fogalmát, mint az állapotokról az értékekre való leképezést.

Ezek az értelmezési primitívák különösen jól használhatók olyan számítási fogalmak modellezésében, mint a matematikai primitívákkal megvalósított állapot-állapot transzformációk. Így a speciális nyelvi konstrukcióknak van egy műveleti szempontból érdekes tulajdonsága és ezek általában könnyűszerrel képezhetők le műveleti specifikációkra.

A szemantikai modellezés területén folyó kutatások részét képezik a specifikus nyelvek definiálására szolgáló szemantikai modellek használatával kapcsolatos vizsgálatok és a meta szintű kutatások. Az utóbbiak a különböző szemantikus tételekben a jelentés különböző fogalmai közötti kapcsolattal foglalkoznak. Speciális nyelveket definiáltak műveleti [109. 110], denotációs [114, 117] és axiomatikus [85, 86] módon. Az axiomatikus specifikációk a nyelvi sajátosságok egy direkt absztrakt jellemzését adják, míg a műveleti és a denotációs specifikációk a nyelvi sajátosságokat azok dinamikus viselkedése szerint jellemzik. A meta szintű kutatásokhoz tartoznak a komplementáris szemantikus specifikációkkal kapcsolatos munkák [115, 118] és a megalapozó munkák, mint amilyenek SCOTT [112, 119] munkái.

A programozási nyelvek szemantikus definíciói általában igen összetett feladatok. A strukturált szemantikára vonatkozó módszertannak a kifejlesztése és a strukturált modell építés nagyon fontos cél, amely párhuzamosan folyik a strukturált programozás hasonló feladatainak megoldásával. Az elméleti kutatás területén egyik alapvető cél, épp úgy, mint az alkalmazási programozásban a komplexitás szervezése és kézben tartása.

IRODALOM

1. *Bevezetés*

[1] Conference on Language Design for Reliable Software, SIGPLAN Notices, March 1977 and CACM, August 1977.
[2] IFIP Working Conference on Formal Description of Programming Concepts, August 1977; ed. E. Neuhold, North-Holland, 1978.
[3] Proceedings of the Symposium on Artificial Intelligence and Programming Languages, SIGPLAN Notices, August 1977.
[4] Design and Implementation of Programming Languages, Proceedings of a DOD-sponsored workshop, Lecture Notes in Computer Science 54, Springer-Verlag, 1977.
[5] Proceedings of the 5th Conference on the Principles of Programming Languages, January. 1978.
[6] Information Processing 77, ed. B. Gilchrist, North-Holland, 1977.
[7] Procedings of the 5th International Conference on Artifical Intelligence, August 1977.
[8] SIAM Journal of Computing, Special Issue on Programming Language Semantics, September 1976.
[9] Ironman: Department of Defense Requirements for Higher Level Programming Languages, SIGPLAN Notices, December 1977.
[10] POPEK, G. J., et al., Notes on the Design of Euclid, in [1], 1977.
[11] GRIES, D., Current Ideas on Programming Methodology, in Research Directions in Software Technology, MIT Press, 1978.
[12] C. W. BACHMAN, The Programmer as Navigator, 1973 Turing Lecture, CACM, November 1973.
[13] J. MCCARTHY, Towards a Mathematical Theory of Computation, IFIP Congress, 1962.
[14] W. H. BURGE, Recursive Programming Techniques, Addison Wesley, 1975.

[15] J. BACKUS, Can Programming be Liberated from the Von Neumann Style? A Functional Style and its Algebra of Programs, CACM, August 1978.
[16] D. E. KNUTH, The Art of Computer Programming, Vol. 3, Sorting and Searching, Addison Wesley, 1972.

2. *A programozási nyelvek fejlődése*

[17] J. SAMMET, Programming Languages, History and Fundamentals, Englewood Cliffs, N. J., Prentice-Hall, 1969.
[18] WEGNER, P., Programming Languages — the first 25 years, IEEE Transactions on Computing, December 1976.
[19] FORTRAN vs. basic FORTRAN, CACM, October 1964.
[20] P. NAUR, ed., Report on the algorithmic language ALGOL 60, CACM May 1960; Revised report on the algortihmic language ALGOL 60, CACM January 1963.
[21] COBOL 1961: Revised specifications for a common business oriented programming language, U.S. Govt. Printing Office, 1961.
[22] J. MCCARTHY et al., LISP 1.5 Programmers Manual, Cambridge, Mass., MIT Press, 1965.
[23] PL/I, Current IBM System 360 Reference Manual or Bates and Douglas, 2nd edition, Englewood Cliffs, N. J., Prentice-Hall, 1975.
[24] A. VAN WIJNGAARDEN et al., Revised report on the algorithmic language ALGOL 68, Acta Informatica 5, pp. 1-236; see also SIGPLAN Notices, May 1977.
[25] A. S. TANENBAUM, A tutorial on Algol 68, Computing Surveys, June 1976.
[26] R. GRISWOLD, J. POAGE and I. POLONSKY, The SNOBOL 4 Programming Language, Englewood Cliffs, N. J., Prentice-Hall, 1971.
[27] D. DAHL and C.A.R. HOARE, Hierarchical program structures, in Dahl, Dijkstra and Hoare, Structured Programming, New York, Academic Press, 1972.
[28] N. WIRTH, The programming language PASCAL, Acta Informatica, 1971, Revised report by K. Jensen and N. Wirth, Springer Verlag, 1974.
[29] A. S. TANENBAUM, A Comparison of Pascal and Algol 68, Report IR—18, Free University of Amsterdam, May 1977. To be published.
[30] L. GILMAN and A. J. ROSE, APL — An interactive approach, 2nd edition, New York, Wiley, 1974.
[31] KEMENY, J. G. and KURTZ, T. E., Basic programming, John Wiley et Sons, New York, 1967.

3. *Nyelvi fogalmak*

3.1. *Típusok*

[32] HOARE, C.A.R., Notes on data structuring, in Structured Programming, Academic Press, New York, 1977.
[33] J. C. REYNOLDS, Gedanken — a simple typeless language based on the principle of completeness and the reference concept, CACM, May 1970.
[34] HABERMAN, N., Critical comments on the language Pascal, Acta Informatica, vol. 3, 1973, pp. 47—58.
[35] GESCHKE, C. M., J. H. MORRIS and E. H. SATHERTHWAITE Early experience with Mesa, CACM, August 1977.

3.2. *Vezérlési struktúrák*

[36] D. E. KNUTH, Structured programming with go to statements, Computing Surveys, December 1974.
[37] WULF, W. A., Programming without the goto, IFIP Congress, 1971.
[38] W. A. WULF, D. B. RUSSELL and A. N. HABERMAN, Bliss — A Language for System Programming, CACM, December 1971.
[39] C. BOHM and G. JACOPINI, Flow diagrams, Turing machines, and languages with only two formation rules, CACM, May 1966.

3.4. *Modularitás, beburkolás, adatabsztrakció*

[40] INGALLS, J., The Smalltalk-76 Programming System, in [5], 1978.
[41] D. L. PARNAS, A technique for software module specification with examples, CACM, May 1972.
[42] LISKOV, B. H. and ZILLES, S. N., Programming with abstract data types, SIGPLAN Notices, April 1974.
[43] WULF, W. A., LONDON, R. and SHAW, M., An introduction to the construction and verification of Alphard programs, IEEE Transactions on Software Engineering, SE—2, 1976, 253—264.
[44] LISKOV, B. H., SNYDER, A., ATKINSON, R. and SHAFFERT, C., Abstraction mechanisms in CLU, CACM, August 1977.

3.5. *Adatabsztrakciók specifikációja*

[45] SHAW, M., WULF, W. A., and LONDON, R. L., Abstraction and verification in Alphard: defining and specifying iteration and generation, CACM, August 1977.
[46] HOARE, C. A. R., Proof of Correctness of Data Representations, Acta Informatica I, 4, 1972.
[47] GUTTAG, J. V., HOROWITZ, E., and MUSSER, D. R., Abstract data types and software validation, ISI/RR—76—48, August 1976.

3.6. *A konkurrens (párhuzamos) feldolgozás absztrakciója*

[48] DIJKSTRA, E. W., Cooperating Sequential Processes, Programming Languages, Ed. F. Genuys, Academic Press, 1968.
[þ9] BRINCH HANSEN, P., The architecture of concurrent programs, Prentice-Hall, 1977.
[50] HOARE, C. A. R., Monitors, an operating system structuring concept, CACM, October 1974.
[51] HOWARD, J. H., Proving monitors, CACM, October 1976.
[52] OWICKI, S., Verifying concurrent programs with shared data classes, in [2], August 1977.
[53] HABERMAN, A. N., Introduction in Operating System Design, Science Research Associates, 1976 [p. 352 for path expressions].

4.2. *Kutatási irányok*

[54] BRYANT, R. and DENNIS, J. B., Concurrent programming, in: *Research Directions in Programming Methodology,* MIT Press, 1979.
[55] REYNOLDS, J., Syntactic Control of Interference, in [5], 1978.
[56] J. JOHNSTON, The contour model of block structured processes, Data Structures in Programming Languages, February 1971.
[57] WEGNER, P., Data structure models in programming languages, SIGPLAN Notices, February 1971.
[58] HAMMER, M., Research directions in data base management, in Research Directions in Software Technology, MIT Press, 1978.
[59] D. G. BOBROW and B. WEGBREIT, A model and stack implementation of multiple environments, CACM, October 1973.
[60] E. W. DIJKSTRA, A discipline of programming, Englewood Cliffs, N. J., Prentice-Hall, 1976.
[61] FLOYD, R. W., Nondeterministic algorithms, JACM, 1967.
[62] BERRY, D. M., and SORKIN, A., The time required for garbage collection in retention block structure languages, Int. J. of Comp. and Sys. Sci. 7, 3 (1978).
[63] SIGPLAN Symposium on very high level languages, SIGPLAN Notices, April 1974.
[64] WIRTH, N., Toward a discipline of real-time programming, CACM, August 1977.
[65] M. HAMMER and G. RUTH, Automatic Programming, in: *Research Directions in Programming Methodology,* MIT Press, 1979.
[66] J. T. SCHWARTZ, Optimizations of Very High Level Languages I, Value and its Corollaries, Computer Languages I, Pergamon Press, 1975.
[67] D. CHAMBERLIN, Relational Data Base Management Systems, special issue on state base management systems, Computing Surveys, March 1976.
[68] D. G. BOBROW and B. RAPHAEL, New Programming Languages for Artificial Intelligence, Computing Surveys, April 1974.

[69] J. B. GOODENOUGH, Exception Handling — Issues and a Proposed Notation, CACM, December 1975.
[70] B. LISKOV and V. BERZINS, Structures Exception Handling, Computation Structures Group Memo 155, December 1977.
[71] R. LEVIN, Program Structures for Exception Condition Handling, PhD Thesis, Carnegie-Mellon University, June 1977.
[72] C. MONTANEGRO, G. Pacini and F. Tuvini, Two Level Control Structure for Non-Deterministic Programming, CACM, October 1977.
[73] D. HAREL and V. R. PRATT, Nondeterminism in logics of programs, in [5], 1978.
[74] D. G. BOBROW and T. WINOGRAD, An overview of KRL — a knowledge representation language, Cognitive Science V, I, no. 1, 1977.

4.3. *Elméleti kutatások*

[75] MANNA, Z., Mathematical theory of computation, McGraw-Hill, 1974.
[76] AHO, A. V. and ULLMAN, J. R., The theory of parsing, translation and compiling, Prentice-Hall, 1972.
[77] M. MARCOTTY, H. F. LEDGAND and G. V. BOCHMANN, A Sampler of Formal Definitions, Computing Surveys, June 1976.

4.3.2. *Objektumokra és tevékenységekre vonatkozó specifikációs sémák*

[78] LISKOV, B. H. and BERZINS, V., An appraisal of program specifications, in: *Research Directions in Programming Methodology*, MIT Press, 1979.
[79] FLOYD, R. W., Assigning meanings to programs, Proc. Symp. Appl. Math. XIX, AMS, 1967.
[80] HOARE, C. A. R., An axiomatic basis for computer programming, CACM, October 1969.
[81] HOARE, C. A. R., Procedures with parameters: an axiomatic approach, in Semantics of algorithmic languages, ed. E. Engeler, Springer-Verlag, 1971.
[82] GUTTAG, J. V., et. al., A proof rule for Euclid procedures, in [2], August 1977.
[83] HOARE, C. A. R., A note on the for statement, BIT 1972.
[84] CLINT, M., and C. A. R. HOARE, Proving jumps and functions, Acta Informatica 1, 1972, 215—224.
[85] HOARE, C. A. R., and WIRTH, N., An axiomatic definition of the programming language Pascal, Acta Informatica 2, no. 4, 1973.
[86] LONDON, R. L. et al., Proof rules for the programming language Euclid, May 1977.
[87] GOGUEN, J. V., Abstract errors for abstract data types, in [2], August 1977.
[88] BURSTALL, R. M. and GOGUEN, J. A., Putting theories together to make specifications, Proc. 5th Joint Conference on Artificial Intelligence, August 1977.
[89] S. A. COOK, Axiomatic and Interpretive Semantics for an Algol Fragment, to be published in SIAM Journal of Computing.
[90] E. M. CLARKE, Programming Language Constructs for Which it is Impossible to Obtain Good HOARE-Like Axiom Systems, Proc. 4th POPL Conference, January 1977.
[91] R. J. LIPTON, A Necessary and Sufficient Condition for the Existence of Hoare Logics, Proc. 17th Symposium on the Foundations of Computer Science, October 1977.

4.3.3. *Verifikáció*

[92] SUZUKI, N., Verifying programs by algebraic and logical reduction, Proc. Int. Conf. on Reliable Software, IEEE, October 1975.
[93] IGARASHI, S., R. L. LONDON and D. C. LUCKHAM, Automatic program verification 1: logical basis and its implementation, Acta Informatica 4, 145—182, 1975.
[94] LUCKHAM, D. C., Program verification and verification-oriented programming, Proc. IFIP, 1977.
[95] LONDON, R. L., Program verification, in: *Research Directions in Programming Methodology*, MIT Press, 1979.
[96] CONSTABLE, R. L., A constructive programming logic. Proc. IFIP, 1977.
[97] MANNA, Z., and WALDINGER, R. J., Is „sometimes" better than „always", Proc. 2d Internat. Conf. on Software Engineering, October 1976, 32—39.

[98] PRATT, V. R., Semantical considerations of Floyd-HOARE Logic, 17th FOCS Symposium, 1976.
[99] MILNER, R. et al., A metalanguage for interactive proof in LCF, in [5], 1978.
[100] KATZ, S., and Z. MANNA, A closer look at termination, Acta Informatica, 4, 1975, 333—352.
[101] D. GRIES, An exercise in proving parallel programs correct, CACM, December, 1977.

4.3.4. *Programtranszformáció és programszintézis*

[102] WEGBREIT, B., Goal-directed program transformation, IEEE Trans. on Software Engineering, June 1976.
[103] J. L. CARTER, A case study of a new code generating technique for compilers, CACM, December 1977.
[104] BURNSTALL, R. M., and DARLINGTON, J., Some transformations in developing recursive programs, JACM, January 1977.
[105] KIBLER, D. F., J. M. Neighbors and T. A. Standish, Program manipulation via an efficient production system, in [3], August 1977.
[106] WEGBREIT, B., Mechanical program analysis, CACM, September 1975.
[107] KAM, J. B., and ULLMAN, J. D., Monotone data flow analysis frameworks, Acta Informatica, 1977.
[108] MANNA, Z. and R. WALDINGER, The automatic synthesis of systems of recursive programs, Proc. Joint Conf. on Artificial Intelligence, MIT, August 1977.

4.3.5. *Szemantika*

[109] P. LUCAS and K. Walk, On the formal description of PL/I, Annual Review of Automatic Programming 6, part 3, New York, Pergamon, 1969.
[110] BEKIC, H. et al., A formal definition of a PL/I subset, IBM Vienna, TR 25.139, December 1974.
[111] D. SCOTT and C. STRACHEY, Towards a mathematical semantics for computer languages, PRG 6, Oxford Computing Laboratory, 1971.
[112] SCOTT, D., Logic and programming languages, CACM, September 1977.
[113] STOY, J. E., Denotational semantics — the Scott—Strachey approach to programming language theory, MIT Press, 1978.
[114] MILNE, R. and STRACHEY, C., A Theory of Programming Languages, CACM, August 1976.
[115] DONAHUE, J. E., Complementary definitions of programming language semantics, Springer Lecture Notes in Computer Science, vol. 42, 1976.
[116] R. D. TENNET, The denotational semantics of programming languages, CACM, August 1976.
[117] DE BAKKER, J. W., Semantics and the foundations of program proving. Proc. IFIP. 1977.
[118] HOARE, C. A. R. and LAUER, P. E., Consistent and complementary theories of the semantics of programming languages, Acta Informatica 3, 1974, 135—154.
[119] SCOTT, D., Data types as lattices, SIAM J. Computing, 1976.

FORDÍTOTTA: VARGA LÁSZLÓ
ELTE TTK NUMERIKUS ÉS GÉPI MATEMATIKA TANSZÉK
1088 BUDAPEST, MÚZEUM KRT. 6—8.

Alkalmazott Matematikai Lapok 6 (1980) 000–000

Könyvismertetés

RÉVÉSZ GYÖRGY: *Bevezetés a formális nyelvek elméletébe.* Akadémiai Kiadó, Budapest, 1979, 154 oldal.

A formális nyelvek elmélete a számítástudomány egyik legdinamikusabban fejlődő területe. A témakörnek jelentős alkalmazási háttere van a fordító, ill. értelmező programok tervezésében.

A könyv 9 fejezete lényegében áttekinti a formális nyelvek elméletével kapcsolatos jelentős eredményeket és részletesen tárgyalja a formális nyelvek elméletének az automaták és algoritmusok elméletéhez, valamint az algoritmusok bonyolultsági kérdéseihez való kapcsolatát.

Az első fejezetben a formális nyelv és a generatív nyelvtan fogalmának és a nyelvtanok gyenge ekvivalenciájának bevezetése után a *Chomsky-hierarchia* definiálása történik meg. A másodikban a nyelveken mint halmazokon végzett műveleteket, a nyelvek szorzatát és a *Kleene-csillagot* definiálják, majd a nyelvosztályoknak a reguláris műveletekre vonatkozó zártsági tulajdonságait mutatja meg a szerző.

A harmadik fejezetben megmutatja, hogy a környezetfüggetlen nyelvek *Chomsky-féle normálalakba* írhatók, ennek segítségével meghatározza a levezetési fájukat, definiálja a (bal) lineáris grammatikákat és a környezetfüggetlen grammatikák *Greibach-féle normálalakját.*

A negyedik fejezet bizonyítja a hosszúságot nem csökkentő grammatikák beágyazottságát a környezetfüggőkbe, bevezeti az előbbiek *Kuroda-féle normálalakját* és az egyoldalú környezetfüggő grammatikákat.

Az ötödik a 0-típusú vagy mondatszerkezetű nyelvekkel foglalkozik, ezekre megad egy normálalakot és a nyelvtan egy eleméhez a levezetési gráfját definiálja.

A hatodik fejezet definiálja a véges automatát 0, 1 vagy 2 veremmel és a *Turing-gépet,* majd az általuk generált nyelvosztályokat veti össze a *Chomsky-hierarchia* osztályaival. Bizonyítja a kétvermes automata és a *Turing-gép* ekvivalenciájának egyik irányát.

A hetedik fejezet definiálja a rekurzív nyelveket, megemlíti, hogy van rekurzíve felsorolható, de nem rekurzív nyelv, ismerteti a *Church-tézist,* majd néhány eldönthetetlen problémát, így a *Turing-gépek* megállási problémáját, az 1-típusú nyelvek ürességének, végtelenségének és két 2-típusú nyelv metszete ürességének problémáját.

A nyolcadik tárgyalja az egyszalagos determinisztikus *Turing-gép* ekvivalenciáját a több szalagos indeterminisztikussal, a gép bonyolultságára mérőszámokat vezet be, az ekvivalens gép bonyolultságára korlátot ad meg, majd a levezetési fát és a felismerési mátrixot ismerteti.

A kilencedik fejezet a szintaktikai elemzés módszereivel foglalkozik; definiálja egy levezetés, ill. egy nyelv egyértelműségét, bevezeti az *Early-féle algoritmust* és egyes grammatikaosztályok szintaktikai analízisének módszereit vizsgálja.

A könyv bizonyításai heurisztikusan vannak megfogalmazva, a halmazelméleti alapismereteket függelék tartalmazza. A könyv szerkesztési módja lehetővé teszi, hogy az olvasó döntse el, hogy milyen mélységben kíván megismerkedni a formális nyelvek elméletével. Ezért matematikusok, mérnökök, kutatók és egyetemi hallgatók egyaránt hasznosan tudják felhasználni a könyvet alapvető definíciók és tételek megismeréséhez.

A könyv irodalomjegyzéke hasznos útmutatást nyújt azoknak a kutatóknak, akik a formális nyelvek elméletének egy speciális részterületén kívánnak elméleti munkásságot kifejteni.

Külön ki kell emelni a kitűzött feladatokat és a kidolgozott példákat az első és a második fejezet végén, melyek nemcsak a tárgyalás szerves részét képezik, hanem alkalmazásokat is szemléltetnek.

DEMETROVICS JÁNOS

A kiadásért felel az Akadémiai Kiadó igazgatója
Műszaki szerkesztő: Sándor István
A kézirat nyomdába érkezett: 1980. október 10. — Terjedelem: 14,90 (A/5 ív)
80-4274 — Szegedi Nyomda — F. v.: 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, főszerkesztő, MTA SZTAKI
1502 Budapest, 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: 50,— Ft

INDEX: 26 498
ISSN 0133—3399

TARTALOMJEGYZÉK

INDEX

Alkalmazott matematikai lapok

1980/3-4

A MAGYAR TUDOMÁNYOS AKADÉMIA
MATEMATIKAI ÉS FIZIKAI TUDOMÁNYOK
OSZTÁLYÁNAK KÖZLEMÉNYEI

6.

KÖTET

AKADÉMIAI KIADÓ, BUDAPEST

ALKALMAZOTT MATEMATIKAI LAPOK

A MAGYAR TUDOMÁNYOS AKADÉMIA MATEMATIKAI ÉS FIZIKAI TUDOMÁNYOK OSZTÁLYÁNAK KÖZLEMÉNYEI

VI. 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.

A kéziratok a főszerkesztőhöz, vagy a szerkesztő bizottság bármely tagjához beküldhetők. A főszerkesztő címe:

Prékopa András, főszerkesztő
1502 Budapest, Kende u. 13—17.

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 100 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 6 (1980) 215—223

EGY ÚJ SZTOCHASZTIKUS KVÁZI-NEWTON MÓDSZER ASZIMPTOTIKUS VIZSGÁLATA

GERENCSÉR LÁSZLÓ

Budapest

Egy $f(\theta)=0$ többváltozós regressziós egyenlet megoldására olyan sztochasztikus approximációs eljárást ismertetünk, amely a *Newton-módszer* sztochasztikus megfelelője. SAKRISON hasonló eredményét annyiban fejlesztjük tovább, hogy az $f_\theta(\theta)$ derivált pontos értéke helyett elég annak egy torzítatlan becslését ismerni. Ha ezen felül a második deriváltat is ismerjük, akkor az új módszer aszimptotikusan efficiens.

1. Bevezetés

Legyen adott egy f: $R^n \to R^1$ függvény és egy

$$f(\theta) = 0 \tag{1.1}$$

többváltozós nemlineáris algebrai egyenlet. Az egyenlet egy gyökét jelölje θ^*. Az (1.1) egyenlet bal oldalát pontosan nem ismerjük, de $f(\theta)$ előállítható $f(\theta)=E\,\xi(\theta)$ alakban, ahol $\xi(\theta)$ egy realizálható valószínűségi változó. Itt E a várható érték képzését jelöli.

θ^* meghatározására jólismert a ROBBINS és MONRO által javasolt

$$\theta(t+1) = \theta(t) + \frac{a}{t}\,\xi(t, \theta(t)) \tag{1.2}$$

algoritmus. Itt a egy pozitív konstans, a $\xi(t, \theta(t))$ valószínűségi változókról pedig a következőt tesszük fel:

$$\xi(t, \theta(t)) = f(\theta(t)) + \varepsilon(t, \theta(t)), \tag{1.3}$$

ahol

$$\varepsilon(t) = \varepsilon(t, \theta(t)) = h(t, \theta(t), \quad e(t)), \tag{1.4}$$

$e(t)$ független a

$$\theta(t), \ldots, \theta(1), \quad e(t-1), \ldots, e(1)$$

változók együttesétől

$$E\{\varepsilon(t, \theta)\} = 0. \tag{1.5}$$

Az $f(\theta)$ függvény, ill. $\operatorname{cov}\varepsilon(t)$ kovariancia-mátrix növekedésére vonatkozó néhány további megszorítás mellett megmutatható, hogy $\theta(t)$ 1 valószínűséggel konvergál θ^*-hoz.

A *Robbins—Monro* vagy *RM típusú módszerek* aszimptotikus tulajdonságait CHUNG [1], SACKS [5], FABIAN [2] és KUSHNER [4] vizsgálták. Az aszimptotikus tulajdonságok birtokában új, hatékonyabb módszereket tervezhetünk, amelyek formálisan szintén *Robbins—Monro típusú módszereknek* tekinthetők.

Ez a törekvés nem új keletű, SAKRISON már 1966-ban javasolt ilyen módszert a *rekurzív maximum likelihood* módszerrel kapcsolatban ([6]). SAKRISON lényegében azzal a feltevéssel él, hogy a regressziós függvény deriváltját ismerjük, és egyfajta *sztochasztikus Newton-módszert* dolgozott ki.

Ebben a dolgozatban egy [3]-ban javasolt *sztochasztikus kvázi-Newton módszer* aszimptotikus tulajdonságait vizsgáljuk és megmutatjuk, hogy a módszer aszimptotikusan efficiens. Az általunk javasolt módszerben elegendő, ha az $f(\boldsymbol{\theta})$ regressziós függvény $f_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ deriváltjának egy torzítatlan becslését ismerjük, továbbá ismerjük az $f_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}^*)$ deriváltakat a $\boldsymbol{\theta}^*$ gyökhelyen. Ez utóbbi feltevés elég erős megszorításnak tűnhet, azonban számos olyan gyakorlati feladat van, ahol a szükséges deriváltakat közelítőleg ki tudjuk számítani. Egyébként a pontos érték ismerete híján is konvergál az algoritmus, ha nem is a lehető leggyorsabban.

2. Az RM módszerek aszimptotikus viselkedése

A következőkben NEVELSZON és HASZMINSZKIJ [7] alapján ismertetjük az (1.2) sztochasztikus approximációs módszer aszimptotikus tulajdonságait. Az eredmények könnyebb értelmezése végett megadjuk a levezetés vázlatát.

Az (1.2) algoritmussal párhuzamosan tekintsük a *Robbins—Monro folyamat* folytonos alakját:

$$d\boldsymbol{\theta}(t) = \frac{a}{t}\left(f(\boldsymbol{\theta}(t)) + \sum_{r=1}^{k} \boldsymbol{\sigma}_r(t, \boldsymbol{\theta}(t))\, dw_r(t)\right). \tag{2.1}$$

Itt $w_1(t), \ldots, w_k(t)$ *standard Wiener folyamatokat* jelölnek, $\boldsymbol{\theta}, \boldsymbol{\sigma}_r(t, \boldsymbol{\theta}(t))$ pedig p dimenziós vektorok.

Tegyük fel, hogy

$$\mathbf{f}(\boldsymbol{\theta}) = \mathbf{B}(\boldsymbol{\theta}-\boldsymbol{\theta}^*)+\delta(\boldsymbol{\theta}-\boldsymbol{\theta}^*) = \mathbf{B}(\boldsymbol{\theta}-\boldsymbol{\theta}^*)+o(\boldsymbol{\theta}-\boldsymbol{\theta}^*), \tag{2.3}$$

ahol $\mathbf{B}$ az $\mathbf{f}(\boldsymbol{\theta})$ függvény deriváltja, és vizsgáljuk a

$$\mathbf{v}(t) = \sqrt{t}\left(\boldsymbol{\theta}(t)-\boldsymbol{\theta}^*\right) \tag{2.4}$$

folyamatot. Ez a folyamat a következő sztochasztikus differenciálegyenlet megoldása:

$$d\mathbf{v}(t) = \frac{\mathbf{A}}{t}\mathbf{v}(t)dt + a\sum_{r=1}^{k} \boldsymbol{\sigma}_r(t, \boldsymbol{\theta}(t))\, dw_r(t) + \frac{a}{\sqrt{t}}\delta\boldsymbol{\theta}(t)\, dt, \tag{2.5}$$

ahol

$$\mathbf{A} = a\mathbf{B}+\frac{1}{2}\mathbf{I} \tag{2.6}$$

és $\mathbf{I}$ az egységmátrix.

A (2.5) egyenlet formálisan tekinthető úgy, mint egy $\mathbf{v}$-re vonatkozó inhomogén lineáris differenciálegyenlet. A differenciálegyenlet együtthatómátrixa $\mathbf{A}/t$, magfüggvénye

$$\boldsymbol{\Phi}(t, u) = \frac{1}{\sqrt{u}} e^{\mathbf{A}\ln t/u}. \tag{2.7}$$

Az (s, t) intervallumon értelmezett megoldás véletlentől függő része előáll

$$\mathbf{v}_0^r(t) = a \int_s^t \mathbf{\Phi}(t, u) \sum_{r=1}^{k} \boldsymbol{\sigma}_r(u, \boldsymbol{\theta}(u))\, dw_r(u) \tag{2.8}$$

alakban $(0<s<t)$.

Tegyük fel, hogy $u \to \infty$ esetén létezik $\boldsymbol{\sigma}_r(u, \boldsymbol{\theta}(u))$ határértéke, és ezt jelöljük $\boldsymbol{\sigma}_r$-rel. Ekkor $\mathbf{v}_0^r(t)$ közelítőleg a következő alakban írható:

$$v_0^r(t) \approx \mathbf{v}_0(t) = a \int_s^t \mathbf{\Phi}(t, u) \sum_{r=1}^{k} \boldsymbol{\sigma}_r\, dw_r(u). \tag{2.9}$$

Egy, az *Ito-integrálokra* vonatkozó fontos azonosság alapján

$$E(\mathbf{v}_0(t)\mathbf{v}_0^T(t)) = a^2 \int_s^t \mathbf{\Phi}(t, u) \left(\sum_{r=1}^{k} \boldsymbol{\sigma}_r \boldsymbol{\sigma}_r^T \right) \mathbf{\Phi}^T(t, u)\, du = \tag{2.10}$$

$$= a^2 \int_s^t \frac{1}{u} e^{\mathbf{A} \ln t/u} \mathbf{S}_0 e^{\mathbf{A}^T \ln t/u}\, du = a^2 \int_0^{\ln t/s} e^{\mathbf{A}v} \mathbf{S}_0 e^{\mathbf{A}^T v}\, dv,$$

ahol

$$\mathbf{S}_0 = \sum_{r=1}^{k} \boldsymbol{\sigma}_r \boldsymbol{\sigma}_r^T. \tag{2.11}$$

Innen $t \to \infty$ esetén megkapjuk a $\mathbf{v}_0(t)$ folyamat kovarianciamátrixának aszimptotikus értékét, feltéve, hogy az létezik. Ennek elégséges feltétele az, hogy ha $\mathbf{A}$ stabil, vagyis ha $\mathbf{A}$ valamennyi karakterisztikus értékének a valós része negatív.

Legyen

$$\mathbf{S} = a^2 \int_0^{\infty} e^{\mathbf{A}v} \mathbf{S}_0 e^{\mathbf{A}^T v}\, dv. \tag{2.12}$$

Ekkor azt kapjuk, hogy $\mathbf{v}(t)$ aszimptotikus eloszlása $N(\mathbf{O}, \mathbf{S})$ típusú, képletben kifejezve

$$\mathbf{v}(t) \sim N(\mathbf{O}, \mathbf{S}). \tag{2.13}$$

A $\mathbf{v}(t)$ kovarianciamátrixának (2.12) előállítása alapján megvizsgálhatjuk az a lépéshossz hatását. Vizsgáljuk először az egyváltozós esetet, és legyen

$$f'(0^*) = b \quad \text{és} \quad S_0 = \sigma_0^2 \tag{2.14}$$

Ekkor a (2.12) integrálra

$$S = \frac{a^2 \sigma_0^2}{2ab - 1} \tag{2.15}$$

adódik, amely minimális akkor, ha

$$a = \frac{1}{b}. \tag{2.16}$$

Ekkor

$$S = \frac{\sigma_0^2}{b^2}. \tag{2.17}$$

Innen származik az a törekvés, hogy a-t legalább közelítőleg a (2.16) képlet alapján határozzuk meg.

Vektorparaméter esetén (1.2) helyett tekintsük a

$$\theta(t+1) = \theta(t) + \frac{\mathbf{C}}{t}\xi(t, \theta(t)) \tag{2.18}$$

folyamatot, ahol az a skalár paraméter helyett most egy $\mathbf{C}$ súlymátrixot vezettünk be. A $\mathbf{C}$ mátrixot úgy kell megválasztani, hogy $\mathbf{CB} + \frac{1}{2}\mathbf{I}$ stabil legyen. Ekkor ugyanis az

$$\eta(t, \theta(t)) = \mathbf{C}\xi(t, \theta(t)) \tag{2.19}$$

választással a

$$\theta(t+1) = \theta(t) + \frac{1}{t}\eta(t, \theta(t)) \tag{2.20}$$

folyamatra közvetlenül alkalmazható a (2.12) (2.13) eredmény. Bevezetve az

$$\mathbf{A} = \mathbf{CB} + \frac{1}{2}\mathbf{I} \tag{2.21}$$

jelölést a

$$\nu(t) = \sqrt{t}(\theta(t) - \theta^*) \tag{2.22}$$

valószínűségi változóra a

$$\nu(t) \sim N(\mathbf{O}, \mathbf{S}) \tag{2.23}$$

eredményt kapjuk, ahol most

$$\mathbf{S} = \mathbf{S}(\mathbf{C}) = \int_0^\infty e^{\mathbf{A}v}\mathbf{C}\mathbf{S}_0\mathbf{C}^T e^{\mathbf{A}^T v}\, dv. \tag{2.24}$$

Megmutatható, hogy a

$$\operatorname{tr} \mathbf{S} \to \min \tag{2.25}$$

probléma megoldása $\mathbf{C} = \mathbf{B}^{-1}$.

3. Sztochasztikus kvázi-Newton módszerek aszimptotikus viselkedése

Ebben a szakaszban röviden ismertetjük SAKRISON módszerét ([7]), majd rátérünk az általunk javasolt módszer vizsgálatára.

SAKRISON feltételezi, hogy $\mathbf{f}_\theta(\theta)$ ismert nemszinguláris mátrix és a következő algoritmust javasolja:

Algoritmus

$$\theta(t+1) = \theta(t) - \frac{1}{t+1}\mathbf{f}_\theta^{-1}(\theta(t))\xi(t, \theta(t)). \tag{3.1}$$

Vezessük be az

$$\eta(t) = \eta(t, \theta(t)) = -\mathbf{f}_\theta^{-1}(\theta(t))\xi(t, \theta(t)) \tag{3.2}$$

jelölést. Ekkor az algoritmus

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \frac{1}{t+1}\boldsymbol{\eta}(t, \boldsymbol{\theta}(t)). \tag{3.3}$$

Alkalmazzuk a (3.3) algoritmusra az előző szakasz eredményeit. A

$$\mathbf{g}(\boldsymbol{\theta}(t)) = E\{\boldsymbol{\eta}(t, \boldsymbol{\theta}(t))|\boldsymbol{\theta}(t)\} = -\mathbf{f}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{\theta}(t))\mathbf{f}(\boldsymbol{\theta}(t)) \tag{3.4}$$

jelölés bevezetésével

$$\mathbf{g}_{\boldsymbol{\theta}}(\boldsymbol{\theta}^*) = -\mathbf{I}. \tag{3.5}$$

Innen a 2. szakaszbeli A mátrixra

$$\mathbf{A} = -\frac{1}{2}\mathbf{I} \tag{3.6}$$

adódik.

Számítsuk ki még az $\boldsymbol{\eta}(t) - \mathbf{g}(\boldsymbol{\theta}(t))$ zaj kovarianciamátrixának aszimptotikus értékét.

Tegyük fel, hogy $\boldsymbol{\varepsilon}(t, \boldsymbol{\theta}(t))$ kovarianciamátrixának aszimptotikus értéke $\mathbf{S}_0$, legyen továbbá

$$\mathbf{B} = \mathbf{f}_{\boldsymbol{\theta}}(\boldsymbol{\theta}^*). \tag{3.7}$$

Ekkor az $\boldsymbol{\eta}(t) - \mathbf{g}(\boldsymbol{\theta}(t))$ zaj kovarianciamátrixának aszimptotikus értékére

$$\mathbf{S}_{0\eta} = \mathbf{B}^{-1}\mathbf{S}_0(\mathbf{B}^{-1})^T \tag{3.8}$$

adódik.

Az $\mathbf{v}(t) = \sqrt{t}(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*)$ valószínűségi vektor változó aszimptotikus eloszlására a 2. szakasz eredményei alapján

$$\mathbf{v}(t) \sim N(\mathbf{O}, \mathbf{S}), \tag{3.9}$$

ahol

$$\mathbf{S} = \int_0^\infty e^{-\frac{v}{2}}\mathbf{S}_{0\eta}e^{-\frac{v}{2}}\,dv = \mathbf{B}^{-1}\mathbf{S}_0(\mathbf{B}^{-1})^T.$$

Ha θ skalárváltozó, akkor a

$$S = \frac{\sigma_0^2}{b^2} \tag{3.11}$$

eredményt kapjuk, ahol $\sigma_0^2 = S_0$, $b = f'(\theta^*)$.

A második sztochasztikus *kvázi-Newton módszer* önálló, új eredmény [3]. Ez olyan esetekben alkalmazható, amikor $\mathbf{f}_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ nem ismert ugyan, de bármely $\boldsymbol{\theta}$-ra realizálható egy $\boldsymbol{\Psi}(\boldsymbol{\theta})$ valószínűségi változó, amelyre

$$\mathbf{f}_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = E(\boldsymbol{\Psi}(\boldsymbol{\theta})). \tag{3.12}$$

Az asszimptotikus efficiencia bizonyításához feltesszük továbbá, hogy ismert az $\mathbf{f}_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}^*)$ második deriváltakból alkotott mátrix, mondjuk

$$\mathbf{f}_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}^*) = \mathbf{C}. \tag{3.13}$$

Az algoritmus alapját a *Newton módszer* egy olyan változata képezi, amelyben $\mathbf{f}_\theta(\theta^*)$ lineárisan szerepel. Bevezetünk egy új $\mathbf{Y}$ változót, amely az $\mathbf{f}_\theta^{-1}(\theta)$ mátrix aktuális értékét közelíti. Az algoritmus egyik része a θ értékét, másik része pedig $\mathbf{Y}$ értékét újítja fel. A t-edik lépésben a változók értékeit $\theta(t)$, $\mathbf{Y}(t)$-vel jelölve az algoritmus a következő:

Algoritmus

$$\theta(t+1) = \theta(t) - \mathbf{Y}(t)\mathbf{f}(\theta(t)) = \theta(t) + \delta\theta(t) \tag{3.14}$$

$$\mathbf{Y}(t+1) = \mathbf{Y}(t) - \frac{1}{2}\mathbf{Y}(t)(\mathbf{f}_\theta(\theta(t))\mathbf{Y}(t) - \mathbf{I}) - \frac{1}{2}\mathbf{Y}(t)\mathbf{f}_{\theta\theta}(\theta(t))\,\delta\theta(t)\mathbf{Y}(t) \tag{3.15}$$

Ez az új algoritmus determinisztikus változata. Az algoritmus második részében szereplő $\frac{1}{2}$ tényezőnek a szerepét később fogjuk megmagyarázni.

A (3.14) (3.15) algoritmus alapján felhasználva $\mathbf{f}_\theta(\theta)$ (3.12) becslését és az $\mathbf{f}_{\theta\theta}(\theta^*) = \mathbf{C}$ értéket a következő sztochasztikus approximációs eljárást kapjuk:

Algoritmus

$$\hat{\theta}(t+1) = \hat{\theta}(t) - \frac{1}{t}\hat{\mathbf{Y}}(t)\xi(t, \theta(t)) \tag{3.16}$$

$$\hat{\mathbf{Y}}(t+1) = \hat{\mathbf{Y}}(t) - \frac{1}{2t}\hat{\mathbf{Y}}(t)(\Psi(t, \theta(t))\hat{\mathbf{Y}}(t) - \mathbf{I}) + \tag{3.17}$$

$$+ \frac{1}{2t}\hat{\mathbf{Y}}(t)\mathbf{C}\hat{\mathbf{Y}}(t)\xi(t, \theta(t))\hat{\mathbf{Y}}(t).$$

A [3] dolgozatban megmutattuk, hogy a (3.14) (3.15) algoritmus kielégíti az [7]-ben megfogalmazott stabilitási feltételeket, vagyis igaz a következő

TÉTEL. A

$$\dot{\theta}(t) = -\mathbf{Y}(t)\mathbf{f}(\theta(t)) \tag{3.18}$$

$$\dot{\mathbf{Y}}(t) = -\frac{1}{2}\mathbf{Y}(t)(\mathbf{f}_\theta(\theta(t))\mathbf{Y}(t) - \mathbf{I}) + \tag{3.19}$$

$$+ \frac{1}{2}\mathbf{Y}(t)\mathbf{C}\mathbf{Y}(t)\mathbf{f}(\theta(t))\mathbf{Y}(t)$$

differenciálegyenlet aszimptotikusan stabilis a $(\theta^*, \mathbf{Y}^*)$ pontban, ahol

$$\mathbf{Y}^* = \mathbf{f}_\theta^{-1}(\theta^*). \tag{3.20}$$

Tekintsük a megfelelő (3.18) (3.19) determinisztikus algoritmust, és számítsuk ki a jobb oldal deriváltját a $(\theta^*, \mathbf{Y}^*)$ helyen.

$$\frac{\partial}{\partial \theta}(\mathbf{Y}\mathbf{f}(\theta))^* = (-\mathbf{Y}\mathbf{f}_\theta(\theta))^* = -\mathbf{I} \tag{3.21}$$

$$\frac{\partial}{\partial \mathbf{Y}}(-\mathbf{Y}\mathbf{f}(\theta))^* = (-\mathbf{f}(\theta))^* = \mathbf{0} \tag{3.22}$$

$$\frac{\partial}{\partial \theta}\left(-\frac{1}{2}\mathbf{Y}(\mathbf{f}_\theta(\theta)\mathbf{Y}-\mathbf{I})+\frac{1}{2}\mathbf{Y}\mathbf{C}\mathbf{Y}\mathbf{f}(\theta)\mathbf{Y}\right)^* = \tag{3.23}$$

$$= (-\mathbf{Y}\mathbf{f}_{\theta\theta}(\theta)\mathbf{Y}+\mathbf{Y}\mathbf{C}\mathbf{Y}\mathbf{f}_\theta(\theta)\mathbf{Y})^* = (\mathbf{Y}(\mathbf{C}-\mathbf{f}_{\theta\theta})\mathbf{Y})^* = \mathbf{0}$$

$$\frac{\partial}{\partial \mathbf{Y}}\left(-\frac{1}{2}\mathbf{Y}(\mathbf{f}_\theta(\theta)\mathbf{Y}-\mathbf{I})+\mathbf{Y}\mathbf{C}\mathbf{Y}\mathbf{f}(\theta)\mathbf{Y}\right)^* = \tag{3.24}$$

$$= -\frac{1}{2}(\mathbf{f}_\theta(\theta)\mathbf{Y}+\mathbf{Y}\mathbf{f}_\theta(\theta))^* = -\mathbf{I}.$$

Így tehát a (2.3) (2.6) képlettel definiált $\mathbf{B}$, ill. $\mathbf{A}$ mátrixokra

$$\mathbf{B} = -\mathbf{I} \tag{3.25}$$

$$\mathbf{A} = -\mathbf{I}+\frac{1}{2}\mathbf{I} = -\frac{1}{2}\mathbf{I} \tag{3.26}$$

adódik.

Számítsuk ki az algoritmusban fellépő véletlen hiba kovariancia mátrixát. Legyen

$$\xi(t, \theta(t)) = \mathbf{f}(\theta(t))+\varepsilon(t, \theta(t)) \tag{3.27}$$

és

$$\Psi(t, \theta(t)) = f_\theta(\theta(t))+\delta(t, \theta(t)). \tag{3.28}$$

Az $\varepsilon(t, \theta(t))$ $\delta(t, \theta(t))$ valószínűségi vektorváltozókról feltesszük, hogy egymástól és a múlttól függetlenek.

Legyen

$$\lim_{t\to\infty} E(\varepsilon(t, \theta(t))\,\varepsilon^T(t, \theta(t))) = \varepsilon_0 \tag{3.29}$$

$$\lim_{t\to\infty} E(\delta(t, \theta(t))\,\delta^T(t, \theta(t))) = \Delta_0. \tag{3.30}$$

A (3.16) (3.17) algoritmus jobboldalán fellépő véletlen hibát a $(\theta^*, \mathbf{Y}^*)$ helyen γ-val jelölve,

$$\gamma = \begin{pmatrix}\gamma_1\\ \gamma_2\end{pmatrix}, \quad \gamma_1 = -\mathbf{Y}^*\varepsilon. \tag{3.31}$$

Legyen

(3.32) $$\mathbf{S}_0 = E(\gamma\gamma^T)$$

és particionáljuk $\mathbf{S}_0$-t a $(\boldsymbol{\theta}, \mathbf{Y})$ particiónak megfelelően:

(3.33) $$\mathbf{S}_0 = \begin{pmatrix} \mathbf{S}_{011} & \mathbf{S}_{012} \\ \mathbf{S}_{021} & \mathbf{S}_{022} \end{pmatrix}.$$

Ekkor

(3.34) $$\mathbf{S}_{011} = E(\mathbf{Y}^* \varepsilon\varepsilon^T (\mathbf{Y}^*)^T) = \mathbf{Y}^* \varepsilon_0 (\mathbf{Y}^*)^T.$$

A

(3.35) $$\mathbf{v}_1(t) = \sqrt{t}(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*)$$

(3.36) $$\mathbf{v}_2(t) = \sqrt{t}(\mathbf{Y}(t) - \mathbf{Y}^*)$$

és

(3.37) $$\mathbf{v}(t) = (\mathbf{v}_1(t), \mathbf{v}_2(t))$$

jelöléseket bevezetve tudjuk, hogy

(3.38) $$\mathbf{v}(t) \sim N(\mathbf{O}, \mathbf{S}),$$

ahol

(3.39) $$\mathbf{S} = \int_0^\infty e^{-\frac{1}{2}v} \mathbf{S}_0 e^{-\frac{1}{2}v}\, dv = \mathbf{S}_0.$$

Ezért

(3.40) $$\mathbf{v}_1(t) \sim N(\mathbf{O}, \mathbf{S}_{011}).$$

Összefoglalva tehát a következőket mondhatjuk:

TÉTEL: A (3.16)—(3.17) sztochasztikus kvázi-Newton módszerre a [7]-ben kimondott mellékfeltételek teljesülése esetén

(3.41) $$\sqrt{t}(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*) \sim N(\mathbf{O}, \mathbf{S}_{011}),$$

ahol

$$\mathbf{S}_{011} = \mathbf{f}_\theta^{-1}(\boldsymbol{\theta}^*) \varepsilon_0 (\mathbf{f}_\theta^{-1}(\boldsymbol{\theta}^*))^T.$$

IRODALOM

[1] CHUNG, K. L., "On stochastic approximation methods", *Ann. Math. Stat.* **25** (1954) 463—483.
[2] FABIAN, V., *On Asymptotic Normality in Stochastic Approximation* (Michigan State University Statistical Laboratory Publication, 1967).
[3] GERENCSÉR, L., "On the use of stability theory in the design of algorithms for structured nonlinear optimization problems", *Problems of Control and Information Theory* **7** (1978) 471—482.
[4] KUSHNER, H. J. and HUANG, H., "Rates of convergence for stochastic approximation type algorithms", *SIAM J. on Control and Opt.* (1979).
[5] SACKS, J., "Asymptotic distribution of stochastic approximations", *Ann. Math. Stat.* **29** (1958) 373—405.

[6] SAKRISON, D. T., "Stochastic approximation: A recursive method for solving regression problem", *Advances in Communication Systems Theory and Applications,* edited by Balakrishnan, A. V. **2**, 51—106, New York, London, 1966.

[7] Невелсон, М. Т. и Хасминский, Р. З., *Стохастическая аппроксимация и рекуррентное оценивание* (Наука, Москва, 1972).

(Beérkezett: 1980. február 25.)

GERENCSÉR LÁSZLÓ
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1137 BUDAPEST, VICTOR HUGO U. 18—22.

ASYMPTOTIC PROPERTIES OF A NEW STOCHASTIC QUASI-NEWTON METHOD

L. GERENCSÉR

For the solution of the regression equation $\mathbf{f}(\boldsymbol{\theta})=\mathbf{0}$ a new stochastic *qusi-Newton method* was developed in an earlier paper. It is an extension of SAKRISON's method in the sense that the assumption on exact knowledge of derivatives is replaced by the assumption that unbiased estimates of the derivatives are known. If second derivatives are known, then the new method is asymptotically efficient.

Alkalmazott Matematikai Lapok 6 (1980) 225—235

SZTOCHASZTIKUS KVÁZI-NEWTON MÓDSZEREK EGY OSZTÁLYÁRÓL

GERENCSÉR LÁSZLÓ

Budapest

A dolgozatban egy a maximum likelihood módszerrel analóg általános becslési elvet mutatunk be. A módszernek több rekurzív változatát is kidolgozzuk és bebizonyítjuk azok stabilitását. Az eredményeket nemlineáris regresszióra és rekurzív maximum likelihood módszerre alkalmazzuk.

1. Bevezetés

A dolgozatban olyan sztochasztikus approximációs eljárásokat dolgozunk ki, amelyek a regressziós függvényről nyert statisztikai információkat jól kihasználják. A problémával egy folyamatszabályozási feladat kapcsán találkoztunk, ahol nagyon fontos volt a konvergenciasebesség javítása.

Sztochasztikus approximációs módszerek konvergenciasebességét sok szerző vizsgálta [3], [7], [9]. Ezekből az eredményekből kitűnik, hogy a leghatékonyabb sztochasztikus approximációs módszerek valamiképpen a *Newton-módszer* sztochasztikus megfelelői. Ez az észrevétel motiválta néhány új *sztochasztikus kvázi-Newton módszer* kidolgozását [4], [5].

Ebben a dolgozatban egy olyan *sztochasztikus kvázi-Newton módszert* mutatunk be, amely a *maximum likelihood módszer*, illetve a *rekurzív maximum likelihood módszer* [2], [6] kiterjesztése. A módszer alapgondolata régi keletű. BARTLETT egy közismert dolgozatában a nemlineáris regresszióból kapott becslés normalitását vizsgálta, illetve javította az itt felhasznált gondolatokkal. Az eljárás során fontos szerepe van az ún. véletlen függvényeknek, amit alább definiálunk. Véletlen függvényekre épülő becslési elvek egy erőteljes elemzését adja a [8] dolgozat.

Legyen adott egy $\xi(\omega)$ valószínűségi változó, és képezzük az $x(\theta, \xi(\omega))$ függvényt, ahol θ skalár. Ezt a függvényt véletlen függvénynek nevezzük. Legyen

$$f(\theta) = Ex(\theta, \xi(\omega)). \tag{1.1}$$

Az $f(\theta)=0$ egyenlet gyökét $x(\theta, \xi(\omega))$ alapján szeretnénk becsülni.

Tegyük fel, hogy $x(\theta, \xi(\omega))$ szigorúan monoton növekvő θ-ban minden ω-ra. Ekkor ugyanez áll $f(\theta)$-ra is. Tegyük fel továbbá, hogy az

$$f(\theta) = 0 \tag{1.2}$$

egyenletnek egyetlen θ^* gyöke van. Ezt az

$$x(\theta, \xi(\omega)) = 0 \tag{1.3}$$

egyenlet gyökével becsüljük. Feltesszük, hogy ennek az egyenletnek minden ω mellett egyetlen $\hat{\theta}$ megoldása van.

A $\hat{\theta}$ becslés eloszlásának vizsgálatához tegyük fel, hogy $x(\theta, \xi(\omega))$ normális eloszlású, mondjuk

$$x(\theta, \xi(\omega)) = f(\theta) + \sigma(\theta) \cdot \eta(\omega), \tag{1.4}$$

ahol $\eta(\omega)$ egy standard normális eloszlású valószínűségi változó.

Az (1.2) egyenlettel párhuzamosan tekintsük az

$$f(\theta) - u_\beta \sigma(\theta) = 0 \tag{1.5}$$

$$f(\theta) + u_\beta \sigma(\theta) = 0 \tag{1.6}$$

egyenleteket, ahol u_β a standard normális eloszlás $0 < \beta < \frac{1}{2}$ valószínűséghez tartozó β kvantilisét jelöli. (Nyilván u_β negatív). Tegyük fel, hogy az (1.4), illetve az (1.5) egyenleteknek egyetlen $\theta_1(\beta)$, illetve $\theta_2(\beta)$ gyöke van.

Igaz a következő

1.1. TÉTEL: A fenti feltételek mellett az (1.3) egyenlet $\hat{\theta}$ megoldása $1-2\beta$ valószínűséggel a $(\theta_1(\beta), \theta_2(\beta))$ intervallumba esik.

Bizonyítás: A valószínűségi mértéket P-vel jelölve

$$P\{\theta < \theta_1(\beta)\} = P\{x(\theta_1(\beta), \xi(\omega)) > 0\}. \tag{1.7}$$

Másrészt

$$0 = f(\theta_1(\beta)) - u_\beta \sigma(\theta_1(\beta)) \tag{1.8}$$

miatt (1.7) jobb oldala β-val egyenlő. Ugyanígy

$$P\{\hat{\theta} > \theta_2(\beta)\} = \beta \tag{1.9}$$

innen adódik már az állítás.

A becslés közelítő eloszlását úgy kaphatjuk meg, hogy linearizáljuk (1.4)-et θ^* körül. Így az

$$f'(\theta^*)(\theta - \theta^*) + (\sigma(\theta^*) + \sigma'(\theta^*)(\theta - \theta^*))\eta(\omega) \approx 0 \tag{1.10}$$

egyenletet kapjuk, $\sigma'(\theta^*) = 0$ esetén innen

$$\hat{\theta} - \theta^* \approx -\frac{\sigma(\theta)}{f'(\theta^*)} \eta(\omega) \tag{1.11}$$

adódik. A $\hat{\theta}$ becslés tehát közelítőleg normális eloszlású, a becslés szórásnégyzete közelítőleg $\sigma^2(\theta^*)/f'(\theta^*)^2$. Ez a megfontolás többdimenziós esetre is kiterjeszthető, a kovarianciamátrix közelítő értéke $f_\theta^{-1}(\theta^*) S(\theta^*) f_\theta^{-1}(\theta^*)$ lesz. Itt $S(\theta^*) x(\theta, \xi(\omega))$ véletlen részének a kovarianciamátrixa.

A fenti elv alapján eredményesen vizsgálhatók véletlen együtthatós lineáris regressziós modellek aszimptotikus tulajdonságai. Legyen

$$y(i) = x^T(i)\theta^* + \varepsilon(i), \tag{1.12}$$

ahol az $\varepsilon(i)$ zaj $N(O, \sigma)$ eloszlású és független a múlttól. Vektor-mátrix alakban (1.12)

$$\mathbf{Y} = \mathbf{X}\theta^* + \mathbf{E} \tag{1.13}$$

alakban írható. Könnyű belátni, hogy

$$\mathbf{E}\{\mathbf{X}^T\mathbf{X}\theta^* - \mathbf{X}^T\mathbf{Y}\} = \mathbf{O}, \tag{1.14}$$

Az

$$\mathbf{X}^T\mathbf{X}\theta^* - \mathbf{X}^T\mathbf{Y} = \mathbf{O} \tag{1.15}$$

egyenlet viszont nem más, mint a legkisebb négyzetes (LKN) becsléshez tartozó normálegyenlet. Az LKN becslés aszimptotikus tulajdonságai tehát a fenti eredmények alapján vizsgálhatók. LJUNG és CAINES megmutatta, hogy a $\hat{\theta}$ LKN becslés kovarianciamátrixára

$$\operatorname{Cov}\hat{\theta} \approx \{E\mathbf{X}^T\mathbf{X}\}^{-1}. \tag{1.16}$$

2. Optimális súlyok több megfigyelés esetén

A következőkben azt vizsgáljuk meg, hogyan célszerű több véletlen függvényből kialakítani egy optimális becslést. Legyenek adottak az $x(i, \theta, \xi(i, \omega))$ véletlen függvények $i = 1, \ldots, n$, ahol most x, θ skalár változók.

2.1. DEFINÍCIÓ: Azt mondjuk, hogy az $x(i, \theta, \xi(i, \omega))$ véletlen függvények függetlenek, ha a $\xi(i, \omega)$ valószínűségi változók függetlenek.

A következőkben $x(i, \theta\, \xi(i, \omega))$-val független véletlen függvényeket jelölünk. Legyen

$$f(i, \theta) = Ex(i, \theta, \xi(i, \omega)), \quad i = 1, \ldots, n. \tag{2.1}$$

Tegyük fel, hogy minden i-re azonos θ^*-gal

$$f(i, \theta^*) = 0, \quad i = 1, \ldots, n, \tag{2.2}$$

és legyen

$$\sigma(x(i, \theta^*, \xi(i, \omega))) = \sigma(i), \tag{2.3}$$

$$f'(i, \theta^*) = \alpha(i). \tag{2.4}$$

Képezzük valamilyen $\lambda(i)$ súlyokkal az

$$X(n, \theta, \eta(n, \omega)) = \sum_{i=1}^{n} \lambda(i) x(i, \theta, \xi(i, \omega)) \tag{2.5}$$

véletlen függvényt, ahol $\eta(n, \omega) = (\xi(1, \omega), \ldots, \xi(n, \omega))^T$. Legyen

$$F(\theta) = EX(n, \theta, \eta(n, \omega)) = \sum_{i=1}^{n} \lambda(i) f(i, \theta). \tag{2.6}$$

Becsüljük θ^*-ot az

$$X(n, \theta, \eta(n, \omega)) = 0 \tag{2.7}$$

egyenletből. Azt mondjuk, hogy a $\lambda(i)$ súlyokat optimálisan választottuk, ha $S(\theta^*)/F'(\theta^*)$ minimális, ahol $S(\theta)$ az $X(n, \theta, \eta(n, \omega))$ véletlen függvény szórását jelöli. Ez a definíció csak $\sigma' = 0$ esetén értelmezhető úgy, hogy a $\hat{\theta}$ becslés szórása a lehető legkisebb.

2.1. TÉTEL. A $\lambda(i)$ súlyok optimális értéke

$$\lambda(i) = \mu \frac{\alpha(i)}{\sigma^2(i)}, \tag{2.8}$$

ahol $\mu \neq 0$ tetszőleges konstans.

Bizonyítás: Számítsuk ki az $S^2(\theta^*)/F'(\theta^*)^2$ szórásnégyzetet:

$$S^2(\theta^*) = \sum_{i=1}^{n} \lambda^2(i)\sigma^2(i), \tag{2.9}$$

$$F'(\theta)^2 = \left(\sum_{i=1}^{n} \lambda(i)\alpha(i)\right)^2. \tag{2.10}$$

A kettő hányadosát kell minimalizálni $\lambda(i)$-ben. A számláló és a nevező a $\boldsymbol{\lambda} = (\lambda(1), \ldots, \lambda(n))^T$ vektornak másodfokú homogén függvénye. A nevező értékét ezért előre rögzíthetjük, mondjuk

$$\sum_{i=1}^{n} \lambda(i)\alpha(i) = 1. \tag{2.11}$$

Most már a (2.9) függvényt kell a (2.11) feltételek mellett minimalizálni. A (2.11) feltételhez tartozó *Lagrange-szorzó* legyen μ. Az

$$L(\lambda, \mu) = \sum_{i=1}^{n} \lambda^2(i)\sigma^2(i) + \mu \sum_{i=1}^{n} \lambda(i)\alpha(i) \tag{2.12}$$

Lagrange-függvény deriváltját tegyük 0-val egyenlővé:

$$L_{\lambda(i)}(\lambda, \mu) = 2\lambda(i)\sigma^2(i) + \mu\alpha(i) = 0. \tag{2.13}$$

Innen a kívánt eredmény azonnal adódik.

Megjegyzés: Megjegyezzük, hogy a probléma vektorértékű $\boldsymbol{\theta}$ esetén is értelmezhető. Legyenek adottak az $\mathbf{x}(i, \boldsymbol{\theta}, \xi(i, \omega))$ ($\mathbf{x}$, $\boldsymbol{\theta}$ p dimenziós) véletlen függvények, és legyen $\mathbf{f}(i, \boldsymbol{\theta}) = E\mathbf{x}(i, \boldsymbol{\theta}, \xi(i, \omega))$. Legyen továbbá

$$\mathbf{f}_{\boldsymbol{\theta}}(i, \boldsymbol{\theta}^*) = \mathbf{A}_i, \quad \operatorname{Cov} \mathbf{x}(i, \boldsymbol{\theta}^*, \xi(i, \omega)) = \mathbf{S}_i, \tag{2.14}$$

ahol $\mathbf{A}_i$ nemszinguláris. Az $\mathbf{x}(i, \boldsymbol{\theta}, \xi(i, \omega))$ véletlen függvényekből egy új véletlen függvényt állítunk elő az

$$\mathbf{X}(n, \boldsymbol{\theta}, \eta(n, \omega)) = \sum_{i=1}^{n} \Lambda_i \mathbf{x}(i, \boldsymbol{\theta}, \xi(i, \omega)) \tag{2.15}$$

lineáris kombinációval, ahol Λ_i-k ismeretlen $p \times p$ méretű nemszinguláris mátrixok.

Az $\mathbf{X}(\boldsymbol{\theta}, \eta(n, \omega))$ véletlen függvény jóságát formálisan a

$$\left(\sum_{i=1}^{n} \Lambda_i \mathbf{A}_i\right)^{-1} \left(\sum_{i=1}^{n} \Lambda_i \mathbf{S}_i \Lambda_i^T\right) \left(\sum_{i=1}^{n} \mathbf{A}_i^T \Lambda_i^T\right)^{-1} \tag{2.16}$$

mátrix nyomával mérjük. Feladat annak a $\Lambda_1, \ldots, \Lambda_n$ rendszernek a meghatározása, amely mellett (2.16) nyoma minimális.

Ez egy tisztán algebrai feladat, amelynek egy megoldása

$$\Lambda_i = \mathbf{S}_i^{-1} \mathbf{A}_i. \tag{2.17}$$

Ezt az eredményt SZTANÓ TAMÁS bizonyította.

3. Rekurzív módszerek

Ebben a pontban a $\boldsymbol{\theta}$ paraméter p dimenziós vektorváltozót jelöl. Legyen adott $\mathbf{x}(i, \boldsymbol{\theta}, \xi(i, \omega))$ p dimenziós véletlen függvények egy független sorozata, és legyen

$$\mathbf{X}(n, \boldsymbol{\theta}, \boldsymbol{\eta}(n, \omega)) = \sum_{i=1}^{n} \mathbf{x}(i, \boldsymbol{\theta}, \xi(i, \omega)), \tag{3.1}$$

ahol

$$\boldsymbol{\eta}(n, \omega) = (\xi(1, \omega), \ldots, \xi(n, \omega)). \tag{3.2}$$

Az

$$\mathbf{X}(n, \boldsymbol{\theta}, \boldsymbol{\eta}(n, \omega)) = \mathbf{O} \tag{3.3}$$

egyenlet megoldását jelölje $\boldsymbol{\theta}(n)$. Azt vizsgáljuk, hogyan lehet újabb megfigyelés esetén $\boldsymbol{\theta}(n+1)$-et közvetlenül $\boldsymbol{\theta}(n)$-ből legalább közelítőleg kiszámítani.

A tisztán numerikus kapcsolat feltárása érdekében tekintsük általánosabban az

$$\mathbf{y}(\boldsymbol{\theta}) = \mathbf{O} \tag{3.4}$$

és az

$$\mathbf{y}(\boldsymbol{\theta}) + \delta \mathbf{y}(\boldsymbol{\theta}) = \mathbf{O} \tag{3.5}$$

egyenleteket. A (3.4), illetve (3.5) egyenletek gyökeit jelölje $\boldsymbol{\theta}(0)$, illetve $\boldsymbol{\theta}(1)$. Tekintsük az

$$\mathbf{y}(\boldsymbol{\theta}) + \varepsilon \delta \mathbf{y}(\boldsymbol{\theta}) = \mathbf{O} \tag{3.6}$$

egyenletsereget, ahol $0 \leqq \varepsilon \leqq 1$. Tegyük fel, hogy a (3.6) egyenletnek egyetlen gyöke van, ezt jelöljük $\boldsymbol{\theta}(\varepsilon)$-nal. A $\boldsymbol{\theta}(1)$ értékét a $\boldsymbol{\theta}(0)$ alapján úgy tudjuk becsülni, hogy a $\boldsymbol{\theta}(\varepsilon)$ függvényt *Taylor-sorának* első két tagjával helyettesítjük. Tehát

$$\boldsymbol{\theta}(1) \approx \boldsymbol{\theta}(0) + \boldsymbol{\theta}_\varepsilon(0). \tag{3.7}$$

A jobb oldal kiszámításához deriváljuk a (3.6) egyenletet ε szerint. Feltéve, hogy $\mathbf{y}(\boldsymbol{\theta})$, $\delta(\boldsymbol{\theta})$ folytonosan differenciálható az $\varepsilon = 0$ helyen, az

$$\mathbf{y}_{\boldsymbol{\theta}}(\boldsymbol{\theta}(0)) \boldsymbol{\theta}_\varepsilon(0) + \delta \mathbf{y}(\boldsymbol{\theta}(0)) = \mathbf{O} \tag{3.8}$$

eredményt kapjuk, ahol az alsó index a parciális deriválást jelöli. Innen a (3.7) felújítási képlet a

$$\theta(1) \approx \theta(0) - \mathbf{y}_\theta^{-1}(\theta(0))\delta\mathbf{y}(\theta(0)) \tag{3.9}$$

alakban írható, feltéve, hogy az $\mathbf{y}_\theta$ derivált mátrix invertálható.

Ennek a közelítő rekurzív kapcsolatnak a felhasználásával rekurzív becslési módszereket dolgozunk ki.

A következőkben a homogén esettel foglalkozunk, vagyis feltesszük, hogy

$$E\mathbf{x}(i, \theta, \xi(i, \omega)) = \mathbf{f}(\theta) \tag{3.10}$$

és

$$E\mathbf{x}_\theta(i, \theta, \xi(i, \omega)) = \mathbf{f}_\theta(\theta). \tag{3.11}$$

Feladatunk az

$$\mathbf{f}(\theta) = \mathbf{O} \tag{3.12}$$

egyenlet egy θ^* gyökének a meghatározása.

A θ^* gyök meghatározására alkalmazzuk a (3.8) rekurzív módszert az

$$\begin{aligned} \mathbf{y}(\theta) &= \mathbf{X}(n, \theta, \eta(n, \omega)) \\ \delta\mathbf{y}(\theta) &= \mathbf{x}(n+1, \theta, \xi(n+1, \omega)) \end{aligned} \tag{3.13}$$

megfeleltetéssel. Az $\mathbf{y}_\theta$ derivált mátrixot kétféleképpen közelíthetjük, ennek megfelelően két algoritmust kapunk. Az első algoritmus során feltételezzük, hogy az $\mathbf{f}_\theta$ derivált mátrix explicit alakban ismert. Ekkor az $\mathbf{y}_\theta(\theta)$-t a várható értékével helyettesítjük és a következőt kapjuk:

3.1. *Algoritmus:*

$$\theta(n+1) = \theta(n) - \frac{1}{n}\mathbf{f}_\theta^{-1}(\theta(n))\mathbf{x}(n+1, \theta(n), \xi(n+1, \omega)). \tag{3.14}$$

Általánosságban nem tehetjük fel, hogy $\mathbf{f}_\theta$ explicit formában ismert. Az $\mathbf{y}_\theta(\theta)$ derivált kiszámításánál azonban egyszerűbb a

$$\mathbf{D}(n) = \frac{1}{n-1}\sum_{i=1}^{n-1}\mathbf{x}_\theta(i+1, \theta(i), \xi(i+1, \omega)) \tag{3.15}$$

közelítő érték kiszámítása, mivel ez rekurzívan számolható. Az algoritmus tehát a következő:

3.2. *Algoritmus:*

$$\theta(n+1) = \theta(n) - \frac{1}{n}\mathbf{D}^{-1}(n)\mathbf{x}(n+1, \theta(n), \xi(n+1, \omega)) \tag{3.16}$$

$$\mathbf{D}(n+1) = \mathbf{D}(n) + \frac{1}{n}(\mathbf{x}_\theta(n+1, \theta(n), \xi(n+1, \omega)) - \mathbf{D}(n)). \tag{3.17}$$

4. Stabilitási eredmények

A javasolt algoritmusok konvergenciatulajdonságait a sztochasztikus approximáció elméletének néhány alapvető eredménye alapján vizsgáljuk. Ezeknek az eredményeknek a részletes ismertetése megtalálható a [9]-ben.

Legyen adott egy

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) + \frac{1}{n}\mathbf{Z}(n, \boldsymbol{\theta}(n), \xi(n, \omega)) \tag{4.1}$$

rekurzióval meghatározott sztochasztikus folyamat, ahol $\boldsymbol{\theta}(n)$ egy p-dimenziós vektor, $\mathbf{Z}(n, \boldsymbol{\theta}, \xi)$ ismert függvény, $\xi(n, \omega)$ pedig független valószínűségi változók egy sorozata.

Minden n-re legyen

$$\mathbf{g}(\boldsymbol{\theta}) = E\mathbf{Z}(n, \boldsymbol{\theta}, \xi(n, \omega)). \tag{4.2}$$

A $\mathbf{Z}(n, \boldsymbol{\theta}(n), \xi(n, \omega))$ valószínűségi változó értelmezhető úgy, mint a $\mathbf{g}(\boldsymbol{\theta})$ függvény alkalmas *Monte Carlo becslése*. Ilyenkor $\xi(n, \omega)$ egy valamilyen értelemben standard, pl. programmal generált véletlen szám (vagy vektor) sorozatot jelöl. Az idézett könyv 5. fejezetében pontosan szerepelnek azok a feltételek, amelyek mellett a $\boldsymbol{\theta}(n)$ sorozat 1-valószínűséggel konvergál a $\mathbf{g}(\boldsymbol{\theta})=\mathbf{O}$ egyenlet megoldáshalmazához. Ebből a meglehetősen bonyolult feltételrendszerből egyetlen feltételt emelünk ki (gyengébb formában), amelynek az algoritmusok tervezésénél véleményünk szerint különösen fontos szerepe van.

Stabilitási feltétel: Azt mondjuk, hogy a (4.1) algoritmusra teljesül a stabilitási feltétel a $\mathbf{g}(\boldsymbol{\theta})=\mathbf{O}$ egyenlet egy $\boldsymbol{\theta}^*$ megoldásában, ha a

$$\dot{\boldsymbol{\theta}} = \mathbf{g}(\boldsymbol{\theta}) \tag{4.3}$$

differenciálegyenlet aszimptotikus stabilis $\boldsymbol{\theta}^*$-ban.

A stabilitási feltétel teljesülését LJAPUNOV eredményeire támaszkodva lehet verifikálni, vagyis azt kell majd ellenőrizni konkrét esetekben, hogy a $\mathbf{g}_{\boldsymbol{\theta}}(\boldsymbol{\theta}^*)$ derivált mátrix valamennyi sajátértéke a negatív féltérben van, más szóval, hogy ez a mátrix stabilis.

A 3. pontban leírt algoritmusok vizsgálatakor a stabilitási feltétel verifikálására szorítkozunk.

4.1. TÉTEL. Tegyük fel, hogy $\mathbf{x}(i, \boldsymbol{\theta}, \xi(i, \omega))$ véletlen függvények egy független sorozata, amelyre teljesülnek a (3.10), (3.11) feltételek, továbbá $\mathbf{f}_{\boldsymbol{\theta}}(\boldsymbol{\theta}^*)$ nemszinguláris. Ekkor a (3.14), illetve a (3.16), (3.17) algoritmusra $\boldsymbol{\theta}^*$-ban teljesül a stabilitási feltétel.

Bizonyítás: A (3.14) algoritmusra

$$-E\{\mathbf{f}_0^{-1}(\boldsymbol{\theta}(n))\mathbf{x}(n, \boldsymbol{\theta}(n), \xi(n+1, \omega))|\boldsymbol{\theta}(n)\} = -\mathbf{f}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{\theta}(n))\mathbf{f}(\boldsymbol{\theta}(n)) = \mathbf{g}(\boldsymbol{\theta}(n)), \tag{4.4}$$

továbbá

$$\mathbf{g}_{\boldsymbol{\theta}}(\boldsymbol{\theta}^*) = -\mathbf{I},$$

ahol $\mathbf{I}$ a $p\times p$ méretű egységmátrixot jelöli, tehát a stabilitási feltétel teljesül.

A (3.16), (3.17) algoritmus esetén a $\theta(n)$, $\mathbf{D}(n)$ változók együttesének rögzítése mellett kell a feltételes várható értéket kiszámítani. (3.16)-ból:

$$-E\{\mathbf{D}^{-1}(n)\mathbf{x}(n+1, \theta(n), \xi(n+1, \omega))|\theta(n), \mathbf{D}(n)\} = -\mathbf{D}^{-1}(n)\mathbf{f}(\theta(n)). \tag{4.5}$$

Ami az algoritmus második felét illeti,

$$E\{\mathbf{x}_\theta(n+1, \theta(n), \xi(n+1, \omega))-\mathbf{D}(n)|\theta(n), \mathbf{D}(n))\} = \mathbf{f}_\theta(\theta(n))-\mathbf{D}(n) \tag{4.6}$$

adódik. Az algoritmushoz hozzárendelt differenciálegyenletrendszer tehát

$$\dot{\theta} = -\mathbf{D}^{-1}\mathbf{f}(\theta) \tag{4.7}$$

$$\dot{\mathbf{D}} = \mathbf{f}_\theta(\theta)-\mathbf{D}, \tag{4.8}$$

ez pedig a θ^*, $\mathbf{D}^*(\mathbf{D}^*=\mathbf{f}_\theta(\theta^*))$ pontban aszimptotikusan stabilis, hiszen a jobb oldal *Jacobi-mátrixa* (θ^*, $\mathbf{D}^*$)-ban

$$\begin{pmatrix} -\mathbf{I} & 0 \\ \mathbf{x} & -\mathbf{I} \end{pmatrix} \tag{4.9}$$

szerkezetű. Ezzel a 4.1 tételt bebizonyítottuk.

5. Alkalmazások

Az előző pontok eredményeit most két konkrét példára alkalmazzuk. Az első példa a *rekurzív maximum likelihood módszer,* amelyet a 3. pont eredményei alapján pontosan elemzünk. A második példa a *rekurzív nemlineáris regresszió,* amelyre a korábbi eredményeket csak formálisan alkalmazhatjuk. A precíz konvergenciavizsgálat meglehetősen bonyolult feltételeit és módszereit tárgyalja az [1] könyv.

Legyen adott egy $h(\mathbf{y}, \theta)$ sűrűségfüggvény, amely függ egy θ p-dimenziós vektor paramétertől. Legyen $\mathbf{Y}$ egy vektorértékű valószínűségi változó, amelynek sűrűségfüggvénye $h(\mathbf{y}, \theta^*)$, ahol θ^* ismeretlen. Feladatunk θ^* becslése $\mathbf{Y}$ realizáció alapján.

Legyen $\mathbf{y}(i, \omega)$ $\mathbf{Y}$ független realizációinak egy sorozata ($i=1, \ldots, n$) és vezessük be az

$$\mathbf{x}(i, \theta, \mathbf{y}(i, \omega)) = h_\theta(\mathbf{y}(i, \omega), \theta)/h(\mathbf{y}(i, \omega), \theta) \tag{5.1}$$

jelölést, amely értelmezve van mindenütt, ahol a nevező nem 0. Feltesszük, hogy bármely rögzített θ mellett az $\{\mathbf{y}: h(\mathbf{y}, \theta)=0\}$ halmaz nullmértékű. Ezekkel az $\mathbf{x}(i, \theta, \mathbf{y}(i, \omega))$ függvényekkel fogjuk alkalmazni a 3. pont eredményeit. Legyen

$$E\mathbf{x}(i, \theta, \mathbf{y}(i, \omega)) = \mathbf{f}(\theta). \tag{5.2}$$

Világos, hogy minden i-re ugyanazt a jobb oldalt kapjuk.

Meg kell mutatnunk, hogy

$$\mathbf{f}(\theta^*) = 0. \tag{5.3}$$

Ez bizonyos megszorítások mellett könnyen bizonyítható. Valóban

$$\mathbf{f}(\theta^*) = E\mathbf{x}(i, \theta^*, \mathbf{y}(i, \omega)) = \int h_\theta(\mathbf{y}(i, \omega), \theta^*)/\mathbf{h}(\mathbf{y}(i, \omega), \theta^*) \cdot \tag{5.4}$$

$$\cdot h(\mathbf{y}(i, \omega), \theta^*)\, d\omega = \int h_\theta(\mathbf{y}(i, \omega), \theta^*)\, d\omega.$$

Ez utóbbi integrál az

$$\int h(\mathbf{y}(i, \omega), \boldsymbol{\theta})\, d\omega = 1 \tag{5.5}$$

integrál $\boldsymbol{\theta}$ szerinti differenciálásával és a műveletek (differenciálás, illetve integrálás) sorrendjének a felcserélésével adódik. Ha ez a felcserélés elvégezhető, azonnal adódik a kívánt $\mathbf{f}(\boldsymbol{\theta}^*)=0$ egyenlőség.

A 3. pontot követve vezessük be az

$$\mathbf{X}(n, \boldsymbol{\theta}, \boldsymbol{\eta}(n, \omega)) = \sum_{i=1}^{n} \mathbf{x}(i, \boldsymbol{\theta}, \mathbf{y}(i, \omega)) \tag{5.6}$$

függvényt. Bevezetve az

$$L(\mathbf{y}(1), \ldots, \mathbf{y}(n), \boldsymbol{\theta}) = \log\left(\prod_{i=1}^{n} h(\mathbf{y}(i), \boldsymbol{\theta})\right) \tag{5.7}$$

likelihood függvényt,

$$\mathbf{X}(n, \boldsymbol{\theta}, \boldsymbol{\eta}(n, \omega)) = L_{\boldsymbol{\theta}}(\mathbf{y}(1), \ldots, \mathbf{y}(n), \boldsymbol{\theta}) \tag{5.8}$$

adódik. Végül a *Fischer-féle információs mátrixra* vezessük be az $\mathbf{I}(\boldsymbol{\theta})$ jelölést:

$$\mathbf{I}(\boldsymbol{\theta}) = \mathbf{L}_{\boldsymbol{\theta}}((1), \boldsymbol{\theta})\mathbf{L}_{\boldsymbol{\theta}}^{T}(\mathbf{y}(1), \boldsymbol{\theta}). \tag{5.9}$$

Ismert, hogy alkalmas regularitási feltételek mellett $\boldsymbol{\theta}=\boldsymbol{\theta}^*$ esetén

$$E\mathbf{L}_{\boldsymbol{\theta}\boldsymbol{\theta}}(\mathbf{y}(1), \ldots, \mathbf{y}(n), \boldsymbol{\theta}^*) = -n\mathbf{I}(\boldsymbol{\theta}). \tag{5.10}$$

Ezek után a 3. fejezetbeli 3.1 algoritmus a következő alakot ölti:

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) + \frac{1}{n\mathbf{I}(\boldsymbol{\theta}(n))} h_{\boldsymbol{\theta}}(\mathbf{y}(n+1, \omega), \boldsymbol{\theta}(n))/h(\mathbf{y}(n+1, \omega), \boldsymbol{\theta}(n)). \tag{5.11}$$

A maximum-likelihood módszernek ez a rekurzív alakja SAKRISONtól származik.

Mivel SAKRISON eredeti dolgozata nehezen hozzáférhető, az érdeklődő olvasó számára NEVELSON és HASZMINSZKIJ könyvét ajánljuk. Itt azt is megmutatják a szerzők, hogy a módszer aszimptotikusan efficiens. Újabb módszereket tartalmaz [2], ill. [6].

A rekurzív maximum-likelihood módszernek egy tudomásunk szerint új változatát kapjuk, ha $\mathbf{I}(\boldsymbol{\theta})$-t becsült értékével helyettesítjük. A (3.16), (3.17) algoritmust kicsit módosított formában alkalmazva, a következőt kapjuk:

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) - \mathbf{C}^{-1}(n) h_{\boldsymbol{\theta}}(\mathbf{y}(n+1, \omega), \boldsymbol{\theta}(n))/h(\mathbf{y}(n+1, \omega), \boldsymbol{\theta}(n)) \tag{5.12}$$

$$\mathbf{C}(n+1) = \mathbf{C}(n) + \mathbf{x}_{\boldsymbol{\theta}}(n+1, \boldsymbol{\theta}(n), \mathbf{y}(n+1, \omega)). \tag{5.13}$$

Itt

$$\mathbf{x}_{\boldsymbol{\theta}}(n+1, \boldsymbol{\theta}, \mathbf{y}(n+1, \omega)) = \frac{\partial^2}{\partial \boldsymbol{\theta}^2} \log h(\mathbf{y}(n+1, \omega), \boldsymbol{\theta}). \tag{5.14}$$

(5.10) figyelembevételével a jobb oldal $\boldsymbol{\theta}=\boldsymbol{\theta}^*$ esetén $-\mathbf{I}(\boldsymbol{\theta}^*)$-gal helyettesíthető. Ezt figyelembevéve (5.13)-at úgy módosíthatjuk, hogy a $h(\mathbf{y}, \boldsymbol{\theta})$ sűrűségfüggvénynek csak az első deriváltjait kelljen használni:

$$\mathbf{C}(n+1) = \tag{5.14}$$

$$= \mathbf{C}(n) - h_{\boldsymbol{\theta}}(\mathbf{y}(n+1, \omega), \boldsymbol{\theta}(n)) h_{\boldsymbol{\theta}}^{T}(\mathbf{y}(n+1, \omega), \boldsymbol{\theta}(n))/h^2(\mathbf{y}(n+1, \omega), \boldsymbol{\theta}(n+1)).$$

Könnyű megmutatni, hogy az algoritmus ebben a formában is kielégíti a stabilitási feltételt θ^*-ban.

Nemlineáris regresszió paraméterének rekurzív becslésére is alkalmazhatók a 3. pont eredményei, hacsak formálisan is. Legyen adott egy

$$y(i, \omega) = g(i, \theta^*) + \varepsilon(i, \omega) \tag{5.15}$$

nemlineáris regresszió, ahol $\varepsilon(i, \omega)$ független standard normális eloszlású valószínűségi változók egy sorozata θ^* ismeretlen p-dimenziós paraméter. Az

$$x\big(i, \theta, y(i, \omega)\big) = y(i, \omega) - g(i, \theta) \tag{5.16}$$

jelölés bevezetésével egy véletlen, skalárértékű függvényt kapunk, amelyre nyilván teljesül az

$$Ex\big(i, \theta^*, y(i, \omega)\big) = 0 \tag{5.17}$$

egyenlőség. A becsléshez az $x\big(i, \theta, y(i, \omega)\big)$ függvények egy alkalmas lineáris kombinációját kell képeznünk. A 2. pont eredményeit formálisan alkalmazva, az i-edik függvényt a

$$\mu_i = g_\theta(i, \theta^*) \tag{5.18}$$

vektorértékű súllyal szorozzuk meg. Ez persze csak elméletileg lehetséges, hiszen θ^* nem ismert. A θ^* becslését tehát elméletileg a

$$\sum_{i=1}^{n} g_\theta(i, \theta^*)\big(y(i, \omega) - g(i, \theta)\big) = 0 \tag{5.19}$$

egyenletből határozzuk meg, gyakorlatilag ehelyett a

$$\sum_{i=1}^{n} g_\theta(i, \theta)\big(y(i, \omega) - g(i, \theta)\big) = 0 \tag{5.20}$$

egyenletet oldjuk meg. Ugyanide jutunk akkor is, ha a legkisebb négyzetes módszert alkalmazzuk.

Rekurzív becslési módszert kaphatunk a 3. pont gondolatainak felhasználásával. A kapott algoritmus az ott közölt két algoritmus keveréke. Explicit formában kapjuk ugyanis a szükséges deriváltakat:

$$Ex_\theta\big(i, \theta, y(i, \omega)\big) = -g_\theta(i, \theta), \tag{5.21}$$

ugyanakkor a deriváltakból alkotott mátrixot célszerű rekurzívan számolni.

A következő algoritmust kapjuk:

$$\theta(n+1) = \theta(n) - \mathbf{C}(n)^{-1} g_\theta\big(n+1, \theta(n), \big(y(n+1, \omega)\big) - g\big(n+1, \theta(n)\big)\big) \tag{5.22}$$

$$\mathbf{C}(n+1) = \mathbf{C}(n) + g_\theta\big(n+1, \theta(n)\big) g_\theta^T\big(n+1, \theta(n)\big). \tag{5.23}$$

Megjegyezzük, hogy az (5.22) és (5.23) algoritmus közvetlenül levezethető úgy is, hogy az (5.15) modellt linearizáljuk és a lineáris modellből rekurzív módon becsüljük θ^*-ot.

IRODALOM

[1] ALBERT, A. E., GARDNER, L. A., *Stochastic approximation and nonlinear regression* (M. I. T. Press, Cambridge, Massachusets, 1967).
[2] BÁNYÁSZ, Cs., KEVICZKY, L., *Discrete time identification of linear dynamic processes* (MTA SZTAKI Tanulmányok, **84**, 1978).
[3] CSIBI, S., *Stochastic processes with learning properties* (Springer Verlag, Wien—New York, 1975).
[4] GERENCSÉR, L., „Egy új sztochasztikus kvázi-Newton módszer aszimptotikus vizsgálata", *Alkalmazott Matematikai Lapok*, 6 (1980).
[5] GERENCSÉR, L., LENGYEL, T., "A derivative-free stochastic quasi-Newton method", MTA SZTAKI working peper.
[6] GERTLER, J., BÁNYÁSZ, Cs., "A recursive (on-line) maximum likelihood identification method", *IEEE Transactions on Automatic Control*, AC **19**, 1974.
[7] KOMLÓS, J., RÉVÉSZ, P., "On the rate of convergence of the RM process", *Zeitschrift Wahrscheinlichkeitstheorie*, **25** (1972) 39—47.
[8] LJUNG, L., CAINES, P. E., "Asymptotic normality of prediction error estimators for approximate system models", *Stochastics* 3 (1979) 29—47.
[9] Невелсон, Р. З., Хасминский, М. Б., *Стохастическая аппроксимация; рекуррентное оценивание*, (Наука, Москва, 1972).

(Beérkezett: 1980. február 25.)

GERENCSÉR LÁSZLÓ
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1137 BUDAPEST, VICTOR HUGO U. 18—22.

ON A CLASS OF STOCHASTIC QUASI-NEWTON METHODS

L. GERENCSÉR

A general estimation principle analogous to the maximum-likelihood principle is developed, together with its recursive variants. Stability properties are proved and applications for nonlinear regression and recursive maximum-likelihood estimation are shown.

Alkalmazott Matematikai Lapok 6 (1980) 237—255

KVADRATIKUS SZTOCHASZTIKUS FELTÉTELLEL BÍRÓ, VALÓSZÍNŰSÉGGEL KORLÁTOZOTT SZTOCHASZTIKUS PROGRAMOZÁSI FELADAT NUMERIKUS MEGOLDÁSA

STUBNYA GUSZTÁVNÉ

Budapest

Ebben a dolgozatban néhány valószínűséggel korlátozott sztochasztikus programozási feladat megoldását ismertetjük. Ezek a feladatok a [4] dolgozatban közölt STABIL sztochasztikus programozási modellnél általánosabb modellből származnak abban az értelemben, hogy a valószínűségi feltételben többváltozós kvadratikus függvény szerepel a korábbi lineáris feltételrendszer helyett. Bár a modellben szereplő valószínűségi változók továbbra is normális eloszlásúak, a fellépő kvadratikus alakok eloszlásfüggvénye általában csak függvénysor alakban állítható elő.

A dolgozat elsődleges célja az, hogy a STABIL modell feladatainál általánosabb sztochasztikus programozási feladatok numerikus megoldására példát mutasson. A közölt numerikus feladatok mind kisméretűek, ezáltal a megoldás menete jól követhető a közölt ábrákon és táblázatokon. Lehetőség nyílik annak megfigyelésére is, hogy hogyan változik a sztochasztikus programozási feladatok optimuma a determinisztikus megfelelőjükhöz képest.

A feladatok megoldására ZOUTENDIJK megengedett irányok módszerét alkalmaztuk, mely a közölt feladatok esetére igen hatékonynak bizonyult. Feltételezhető azonban, hogy nagyobb méretű feladatok hatékony megoldásához szükség lehet más nemlineáris programozási algoritmusok használatára is, amint az a STABIL modell nagyméretű feladata esetén is történt (lásd [6], [7]).

1. Bevezetés

A sztochasztikus programozás döntési modelljei számos gyakorlati feladat matematikai leírására szolgálnak. Ilyen feladatok például a táplálási, szállítási és a tározó rendszerek tervezésével kapcsolatos problémák. A sztochasztikus programozás döntési modelljei közül leggyakrabban a valószínűséggel korlátozott programozást és a kétlépcsős sztochasztikus programozás döntési modelljeit alkalmazzuk. Több ilyen modell megfogalmazása, az elmélet kidolgozása és a megoldó algoritmus megadása PRÉKOPA ANDRÁS nevéhez fűződik. Ilyen például a [4]-ben közölt STABIL elnevezésű valószínűséggel korlátozott sztochasztikus programozási modell is,

$$
\begin{aligned}
(1.1) \qquad & G(\mathbf{x}) = P\big(g_i(\mathbf{x}) \geqq \xi_i,\ i = 1, \ldots, m\big) \geqq p \\
& g_i(\mathbf{x}) \geqq b_i, \qquad i = m+1, \ldots, M \\
& \min f(\mathbf{x}),
\end{aligned}
$$

ahol a $g_1(\mathbf{x}), \ldots, g_{m+M}(\mathbf{x})$, $f(\mathbf{x})$ függvények lineárisak. A modellben a $\xi_1, \ldots, \xi_m$ valószínűségi változók együttes eloszlása folytonos és az együttes sűrűségfüggvény az egész téren logaritmikusan konkáv.

Ebben a dolgozatban az alábbi modellből származó feladatokat oldunk meg:

$$G(\mathbf{x}) = P(g(\mathbf{x}, \xi) \geqq 0) \geqq p \tag{1.2}$$

$$\mathbf{a}_i' \mathbf{x} \geqq b_i, \quad i = 1, \ldots, m$$

$$\min \mathbf{c}' \mathbf{x},$$

ahol

$$g(\mathbf{x}, \xi) = \begin{pmatrix} \mathbf{x} \\ \xi \end{pmatrix}' \mathbf{A} \begin{pmatrix} \mathbf{x} \\ \xi \end{pmatrix} + \mathbf{a}' \begin{pmatrix} \mathbf{x} \\ \xi \end{pmatrix} + b, \tag{1.3}$$

$\mathbf{A}$ $n+m$-edrendű szimmetrikus mátrix, $\mathbf{a} \in R^{n+m}$, $\mathbf{x} \in R^n$, $\xi \in R^m$, $b \in R$ (n és m természetes számok). A ξ valószínűségi vektorváltozó komponenseinek együttes eloszlása folytonos és az együttes sűrűségfüggvényük az egész téren logaritmikusan konkáv.

Ez a modell a STABIL modellnél általánosabb abban az értelemben, hogy a sztochasztikus feltételben többváltozós kvadratikus függvény szerepel a lineáris feltételrendszer helyett. Továbbá egyszerűbb is, mert csak egy függvényt tartalmaz a sztochasztikus feltétel.

Minden feladatban a $\xi_1, \ldots, \xi_m$ valószínűségi változók együttes eloszlása normális eloszlás, tehát az együttes sűrűségfüggvényük logaritmikusan konkáv az egész téren. Feltesszük még, hogy az $\mathbf{A}$ mátrix minden sajátértéke nem-pozitív. Ebből következik, hogy a $g(\mathbf{x}, \mathbf{y})$ függvény konkáv. A [2] dolgozatban bizonyított tétel alapján, a mi esetünkben a $G(\mathbf{x}) = P(g(\mathbf{x}, \xi) \geqq 0)$ függvény logaritmikusan konkáv, tehát kvázikonkáv is.

A feladatok megoldására olyan nemlineáris programozási módszerek alkalmasak, amelyek lineáris célfüggvény és kvázikonkáv feltételi függvények esetén konvergensek. Ebben a dolgozatban ZOUTENDIJK egyik „megengedett irányok" elnevezésű módszerét, az ún. P2 módszert alkalmaztuk (lásd [9]). Ez az algoritmus bizonyos regularitási feltételek teljesülése esetén kvázikonkáv feltételi függvények mellett is konvergens (lásd [1], [4]).

Köszönetet mondok ezúton is DR. PRÉKOPA ANDRÁS egyetemi tanárnak, aki a téma kiválasztásában és kidolgozásában hasznos tanácsaival segített.

2. A sztochasztikus feltétel vizsgálata

Az (1.3) kvadratikus függvénynek csak azokkal a speciális eseteivel foglalkozunk, amikor az $\mathbf{A}$ mátrix az alábbi alakú

$$\mathbf{A} = \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{pmatrix},$$

ahol $\mathbf{B}$ $n \times n$-es, $\mathbf{D}$ $m \times m$-es szimmetrikus mátrix. Az $\mathbf{A}$ mátrix többi eleme 0, és minden sajátértéke nem-pozitív. Ekkor a főtengely-transzformáció végrehajtásával az (1.3) függvény a következő alakot ölti:

$$g(\mathbf{x}, \xi) = \sum_{i=1}^{n} \lambda_i x_i^2 + \sum_{i=1}^{m} \mu_i \xi_i^2 + \sum_{i=1}^{n} \alpha_i x_i + \sum_{j=1}^{m} \alpha_{n+j} \xi_j + \beta, \tag{2.1}$$

ahol $\lambda_i \leqq 0$, $i=1, \ldots, n$, $\mu_j \leqq 0$, $j=1, \ldots, m$ az **A** mátrix sajátértékei; α_i, $i=1, \ldots, n$, α_{n+j}, $j=1, \ldots, m$ az $\mathbf{a} \in R^{n+m}$ vektor komponenseinek megfelelő transzformált értékek és $\beta = b \in R$.

Ha $\mu_i < 0$, $i=1, \ldots, m$, akkor a ξ_i, $i=1, \ldots, m$ valószínűségi változókat tartalmazó tagok összevonásával:

$$g(\mathbf{x}, \xi) = \sum_{i=1}^{m} \mu_i \left(\xi_i + \frac{\alpha_{n+i}}{2\mu_i}\right)^2 + \sum_{i=1}^{n} \lambda_i x_i^2 + \sum_{i=1}^{n} \alpha_i x_i + \beta - \sum_{i=1}^{m} \frac{\alpha_{n+i}^2}{4\mu_i},$$

ha pedig $\mu_i < 0$, $i=1, \ldots, r$ és $\mu_i = 0$, $i=r+1, \ldots, m$, akkor

$$g(\mathbf{x}, \xi) = \sum_{i=1}^{r} \mu_i \left(\xi_i + \frac{\alpha_{n+i}}{2\mu_i}\right)^2 + \sum_{i=r+1}^{m} \alpha_{n+i} \xi_i + \sum_{i=1}^{n} \lambda_i x_i^2 + \sum_{i=1}^{n} \alpha_i x_i + \beta - \sum_{i=1}^{r} \frac{\alpha_{n+i}^2}{4\mu_i}.$$

Ekkor az (1.2) modellben szereplő $g(\mathbf{x}, \xi) \geqq 0$ sztochasztikus feltétel a következő lesz:

$$-\sum_{i=1}^{m} \mu_i \left(\xi_i + \frac{\alpha_{n+i}}{2\mu_i}\right)^2 \leqq u_1(\mathbf{x}), \quad \text{ha} \quad \mu_i < 0, \quad i = 1, \ldots, m,$$

illetve

$$-\sum_{i=1}^{r} \mu_i \left(\xi_i + \frac{\alpha_{n+i}}{2\mu_i}\right)^2 - \sum_{i=r+1}^{m} \alpha_{n+i} \xi_i \leqq u_2(\mathbf{x}),$$

ha $\mu_i < 0$, $i=1, \ldots, r$ és $\mu_i = 0$, $i=r+1, \ldots, m$, ahol

$$u_1(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i x_i^2 + \sum_{i=1}^{n} \alpha_i x_i + \beta - \sum_{i=1}^{m} \frac{\alpha_{n+i}^2}{4\mu_i}$$

$$u_2(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i x_i^2 + \sum_{i=1}^{n} \alpha_i x_i + \beta - \sum_{i=1}^{r} \frac{\alpha_{n+i}^2}{4\mu_i}.$$

A $P(g(\mathbf{x}, \xi) \geqq 0)$ valószínűség számításához tehát az

$$\eta = -\sum_{i=1}^{m} \mu_i \left(\xi_i + \frac{\alpha_{n+i}}{2\mu_i}\right)^2,$$

illetve

$$\eta = -\sum_{i=1}^{r} \mu_i \left(\xi_i + \frac{\alpha_{n+i}}{2\mu_i}\right)^2 - \sum_{i=r+1}^{m} \alpha_{n+i} \xi_i$$

valószínűségi változó sűrűség — és eloszlásfüggvényét kell ismerni, ahol a ξ_i, $i=1, \ldots, m$ valószínűségi változók normális eloszlásúak.

Jelölje $h(y)$, illetve $H(y)$ az η valószínűségi változó sűrűség, illetve eloszlásfüggvényét. Ezek könnyen felírhatók impropius paraméteres integrál alakjában. A paraméteres integrálokra vonatkozó tételek segítségével következtetni lehet e függvények folytonosságára és $H(y)$ differenciálhatóságára. Az integrálás eredményeként, hosszabb számolás után e függvények általában függvénysor alakban adódnak.

A 3. szakaszban közölt mintafeladatok mindegyikében $n=2$, $m=2$ és csak az **A** mátrix megválasztásában, illetve a ξ_1, ξ_2 alap valószínűségi változók normális együttes eloszlásának a paramétereiben különböznek egymástól.

A továbbiakban a helyenként hosszadalmas, de elemi számítások mellőzésével felsoroljuk a mintafeladatokban felhasznált speciális eseteknek megfelelő η valószínűségi változók sűrűség, illetve eloszlásfüggvényeit.

1) Legyen

$$\eta = \xi_1^2 + \xi_2^2,$$

ahol ξ_1, ξ_2 független, normális eloszlású valószínűségi változók, 0 várható értékkel és 1 szórással. Ekkor, mint az jól ismert, η 2-szabadságfokú χ^2 eloszlású és a sűrűségfüggvénye:

$$h(y) = \begin{cases} \frac{1}{2} e^{-\frac{y}{2}}, & \text{ha} \quad y > 0 \\ 0 \quad , & \text{ha} \quad y \leqq 0. \end{cases} \tag{2.2}$$

2) Legyen

$$\eta = \xi_1^2 + \xi_2^2,$$

ahol ξ_1, ξ_2 független, normális eloszlású valószínűségi változók, 0 várható értékkel és $\sigma_1 \neq \sigma_2$ szórással. Ekkor η sűrűségfüggvénye:

$$h(y) = \begin{cases} \frac{1}{2\sqrt{\pi} \cdot \sigma_1 \sigma_2} e^{-\frac{y}{2\sigma_1^2}} \sum_{n=0}^{\infty} \frac{y^n}{n!} \left(\frac{1}{2\sigma_1^2} - \frac{1}{2\sigma_2^2} \right)^n \frac{\Gamma\left(n + \frac{1}{2}\right)}{\Gamma(n+1)}, & \text{ha} \quad y > 0 \\ 0 \quad , & \text{ha} \quad y \leqq 0. \end{cases} \tag{2.3}$$

3) Legyen

$$\eta = \xi_1^2 + \xi_2^2,$$

ahol ξ_1, ξ_2 független, normális eloszlású valószínűségi változók, m_1, m_2 várható értékkel és 1 szórással. Ekkor η sűrűségfüggvénye:

$$h(y) = \begin{cases} \frac{1}{2\sqrt{\pi}} e^{-\frac{y}{2} - \frac{d^2}{2}} \sum_{n=0}^{\infty} \frac{y^n d^{2n}}{(2n)!} \frac{\Gamma\left(n + \frac{1}{2}\right)}{\Gamma(n+1)}, & \text{ha} \quad y > 0 \\ 0 \quad , & \text{ha} \quad y \leqq 0, \end{cases} \tag{2.4}$$

ahol $d^2 = m_1^2 + m_2^2$ (lásd [8]).

4) Legyen

$$\eta = \xi_1^2 + \xi_2^2,$$

ahol ξ_1, ξ_2 nem független, normális eloszlású valószínűségi változók, 0 várható értékkel, 1 szórással és 0,5 korrelációs együtthatóval. Ekkor η eloszlásfüggvénye:

$$H(y) = \begin{cases} \frac{1}{\sqrt{3}} \left\{ y + \sum_{n=2}^{\infty} (-1)^{n+1} \frac{y^n}{n!} \left[1 + \sum_{k=1}^{n-1} \binom{n-1}{k} \left(\frac{-2}{3} \right)^k \frac{(2k-1) \dots 1}{(2k) \dots 2} \right] \right\}, & \text{ha} \quad y > 0 \\ 0 \quad , & \text{ha} \quad y \leqq 0. \end{cases} \tag{2.5}$$

5) Legyen

$$\eta = \xi_1^2 + \xi_2,$$

ahol ξ_1, ξ_2 független, normális eloszlású valószínűségi változók, 0 várható értékkel és 1 szórással. Ekkor η sűrűségfüggvénye:

$$h(y) = \frac{1}{2^{7/4}\pi} e^{-\frac{y^2}{2}} \sum_{n=0}^{\infty} (-1)^n \frac{1}{n!} \left(\frac{1-2y}{\sqrt{2}}\right)^n \Gamma\left(\frac{n}{2}+\frac{1}{4}\right). \tag{2.6}$$

3. Numerikus feladatok megoldása

A numerikus feladatokat ZOUTENDIJK megengedett irányok módszerével (lásd [9] P2 algoritmus) oldottuk meg. Ezt röviden az alábbiakban foglaljuk össze.

Legyen $\mathbf{x}_1$ az (1.2) feladat feltételeit kielégítő vektor. Az egymás utáni vektorokat iterációval határozzuk meg. Tételezzük fel, hogy az $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$ vektorokat már ismerjük. A $k+1$-edik iterációban az $\mathbf{x}_{k+1}$ vektort határozzuk meg. Az iteráció első részében megoldjuk az alábbi iránykereső feladatot:

$$\begin{aligned} G(\mathbf{x}_k)+\nabla G(\mathbf{x}_k)\cdot(\mathbf{x}-\mathbf{x}_k)+\vartheta y &\geqq p \\ \mathbf{a}_i'\mathbf{x} &\geqq b_i, \quad i = 1, \ldots, m \\ \mathbf{c}'(\mathbf{x}-\mathbf{x}_k) &\leqq y \\ \min y, & \end{aligned}$$

ahol ϑ tetszőleges, de az egész eljárás alatt rögzített pozitív szám. Ha $y_{\text{opt}}=0$, akkor az eljárás véget ér. Ha $y_{\text{opt}}<0$, akkor rátérünk az iteráció második részére, a lépéshossz meghatározására.

Jelölje $\mathbf{x}_k^*$ az y_{opt}-hoz tartozó optimális megoldást. Ezután minimalizáljuk a

$$\mathbf{c}'(\mathbf{x}_k+\lambda(\mathbf{x}_k^*-\mathbf{x}_k))$$

függvényt azon λ-k halmazán, melyekre $\lambda \geqq 0$, és $\mathbf{x}_k+\lambda(\mathbf{x}_k^*-\mathbf{x}_k)$ eleget tesz az (1.2) feladat feltételeinek. Jelölje λ_k a minimumot megvalósító helyet. Ekkor $\mathbf{x}_{k+1}$ értelmezése a következő:

$$\mathbf{x}_{k+1} = \mathbf{x}_k+\lambda_k(\mathbf{x}_k^*-\mathbf{x}_k).$$

Ezeket az iterációs lépéseket alkalmazzuk mindaddig, amíg vagy véges sok lépéssel az optimumhoz jutunk; vagy az $\mathbf{x}_k$ vektort már optimális megoldásnak tekintjük, ha az $\mathbf{x}_{k+1}$ és $\mathbf{x}_k$ vektorokhoz tartozó célfüggvényértékek eltérése az $\mathbf{x}_k$ vektorhoz tartozó célfüggvényértéknek legfeljebb 1 %-a.

Az $\mathbf{x}_1$ vektor meghatározásához oldjuk meg először a

$$P(\eta \leqq u) = p$$

egyenletet. Jelölje a megoldást u_0. Ezután megoldjuk az alábbi kvadratikus programozási feladatot:

$$\mathbf{a}_i'\mathbf{x} \geqq b_i, \quad i = 1, \ldots, m$$

$$\max\left(\sum_{i=1}^{n} \lambda_i x_i^2 + \sum_{i=1}^{n} \alpha_i x_i + \beta_1\right),$$

ahol $\lambda_i \leqq 0$, $i=1, \ldots, n$, tehát a célfüggvény konkáv. A megoldást WOLFE-tól szár-

mazó módszerrel végeztük el (lásd [10]). Jelölje $\mathbf{x}_{opt}$ a feladat optimális megoldását és u_{opt} az optimum értékét.

Ha $u_{opt}<u_0$, akkor az (1.2) feladatnak nincs megengedett megoldása. Ha $u_{opt}=u_0$, akkor az (1.2) feladatnak van megengedett megoldása, de a P2 algoritmus konvergenciája nem feltétlen biztosított. Ha $u_{opt}>u_0$, akkor az (1.2) feladatnak van megengedett megoldása, és a P2 algoritmus bizonyítottan konvergens. Ekkor legyen $\mathbf{x}_1=\mathbf{x}_{opt}$, melyre biztosan tudjuk, hogy $G(\mathbf{x}_1)>p$.

A fent ismertetett eljárás konvergens, mert a $G(\mathbf{x})=P(g(\mathbf{x},\xi)\geqq 0)$ függvény kvázikonkáv, folytonosan differenciálható egy konvex halmazon; a célfüggvény lineáris; a lineáris feltételek által meghatározott halmaz korlátos; létezik olyan $\mathbf{z}$ vektor, amely a feltételeket kielégíti és $G(\mathbf{z})>p$.

A feladatokban azokat az eloszlás és sűrűségfüggvényeket, amelyeket függvénysorral adtunk meg, a függvénysor megfelelő indexű részletösszegével közelítjük. Az intervallumot, amelyen a közelítést 10^{-4} nagyságrendű hibával végeztük, az $u(\mathbf{x})$ függvénynek a lineáris feltételek által meghatározott halmazon felvett szélsőértékei határozták meg.

1. Feladat

$$\begin{aligned} P(\eta \leqq 1-x_1^2+x_1+x_2) &\geqq 0{,}8 \\ x_1+2x_2 &\geqq 1 \\ -3x_1-2x_2 &\geqq -6 \\ x_1 &\geqq 0 \\ x_2 &\geqq 0 \\ \min\,(x_1+x_2) & \end{aligned}$$

Az $\eta=\xi_1^2+\xi_2^2$ valószínűségi változó sűrűségfüggvénye (2.2), várható értéke 2, és eloszlásfüggvénye:

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

Az $\mathbf{x}_1=\begin{pmatrix}0{,}5\\2\end{pmatrix}$ vektorra minden feltétel teljesül és $G(\mathbf{x}_1)>0{,}8$. Az iterációk részeredményeit az 1. táblázat mutatja.

Mivel $y_{3opt}=0$, a feladat optimális megoldása

$$\mathbf{x}_3=\begin{pmatrix}0\\2{,}218\,880\,1\end{pmatrix}, \quad \text{és} \quad \min\,(x_1+x_2)=2{,}218\,880\,1.$$

A determinisztikus feladat

$$\begin{aligned} 2 \leqq 1-x_1^2+x_1+x_2 & \\ x_1+2x_2 &\geqq 1 \\ -3x_1-2x_2 &\geqq -6 \\ x_1 &\geqq 0 \\ x_2 &\geqq 0 \\ \min\,(x_1+x_2) & \end{aligned}$$

1. TÁBLÁZAT

Az iterációk részeredményei

i	$\mathbf{x}_i$	$u(\mathbf{x}_i)$	$G(\mathbf{x}_i)$	$h(u(\mathbf{x}_i))$	$(\nabla u(x_i))'$	$f(\mathbf{x}_i)$	$y_{i\,\mathrm{opt}}$	$\mathbf{x}_i^*$
1	$\begin{pmatrix}0{,}5\\2\end{pmatrix}$	3,25	0,803 09	0,098 455	$\begin{pmatrix}0\\1\end{pmatrix}$	2,5	−0,047 608 7	$\begin{pmatrix}0\\2{,}452\,371\,2\end{pmatrix}$
2	$\begin{pmatrix}0\\2{,}452\,371\,2\end{pmatrix}$	3,452 371 2	0,822 04	0,088 98	$\begin{pmatrix}1\\1\end{pmatrix}$	2,452 371 2	−0,020 239 2	$\begin{pmatrix}0\\2{,}432\,131\,8\end{pmatrix}$
3	$\begin{pmatrix}0\\2{,}218\,880\,1\end{pmatrix}$	3.218 88	0,8	0,1	$\begin{pmatrix}1\\1\end{pmatrix}$	2,218 880 1	0	$\begin{pmatrix}0\\2{,}218\,880\,1\end{pmatrix}$

2. TÁBLÁZAT

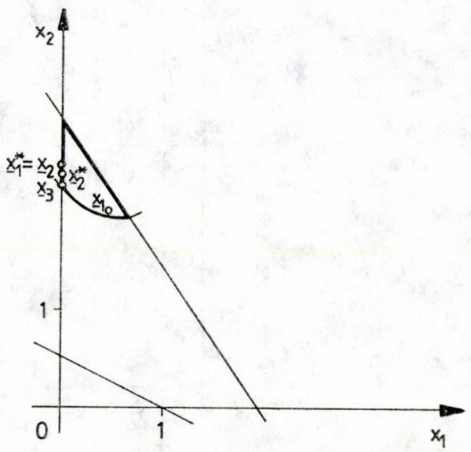
Az iterációk részeredményei

i	$\mathbf{x}_i$	$u(\mathbf{x}_i)$	$G(\mathbf{x}_i)$	$h(u(\mathbf{x}_i))$	$(\nabla u(\mathbf{x}_i))'$	$f(\mathbf{x}_i)$	$y_{i\,\mathrm{opt}}$	$\mathbf{x}_i^*$
1	$\begin{pmatrix}0{,}5\\0{,}5\end{pmatrix}$	4,5	0,894 600 8	0,052 699 6	$\begin{pmatrix}0\\0\end{pmatrix}$	1	−0,094 6	$\begin{pmatrix}0\\0{,}5\end{pmatrix}$
2	$\begin{pmatrix}0\\0{,}5\end{pmatrix}$	4,25	0,880 57	0,059 715	$\begin{pmatrix}1\\0\end{pmatrix}$	0,5	0	$\begin{pmatrix}0\\0{,}5\end{pmatrix}$

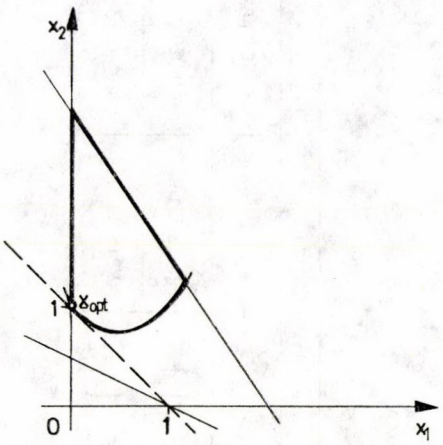
megoldása

$$\mathbf{x}_{opt} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \text{és} \quad \min(x_1 + x_2) = 1.$$

A sztochasztikus és a determinisztikus feladat megoldását az 1. és a 2. ábra szemlélteti. A két feladat optimális megoldása és az optimum értéke különböző.



1. ábra



2. ábra

2. Feladat

$$P(\eta \leqq 4 - x_1^2 - x_2^2 + x_1 + x_2) \geqq 0{,}8$$
$$x_1 + 2x_2 \geqq 1$$
$$-3x_1 - 2x_2 \geqq -6$$
$$x_1 \geqq 0$$
$$x_2 \geqq 0$$
$$\min(x_1 + x_2)$$

Az $\eta = \xi_1^2 + \xi_2^2$ valószínűségi változó sűrűségfüggvénye (2.2), várható értéke 2, és eloszlásfüggvénye:

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

Az $\mathbf{x}_1 = \begin{pmatrix} 0{,}5 \\ 0{,}5 \end{pmatrix}$ vektorra minden feltétel teljesül, és $G(\mathbf{x}_1) > 0{,}8$. Az iterációk részeredményeit a 2. táblázat tartalmazza.

Mivel $y_{opt} = 0$, a feladat optimális megoldása

$$\mathbf{x}_2 = \begin{pmatrix} 0 \\ 0{,}5 \end{pmatrix}, \quad \text{és} \quad \min(x_1 + x_2) = 0{,}5.$$

A determinisztikus feladat

$$2 \leqq 4-x_1^2-x_2^2+x_1+x_2$$

$$x_1+2x_2 \geqq 1$$

$$-3x_1-2x_2 \geqq -6$$

$$x_1 \geqq 0$$

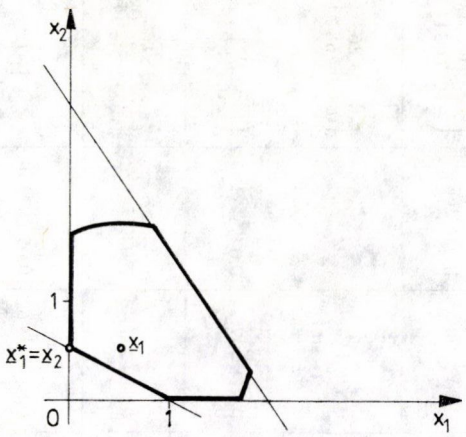
$$x_2 \geqq 0$$

$$\min (x_1+x_2)$$

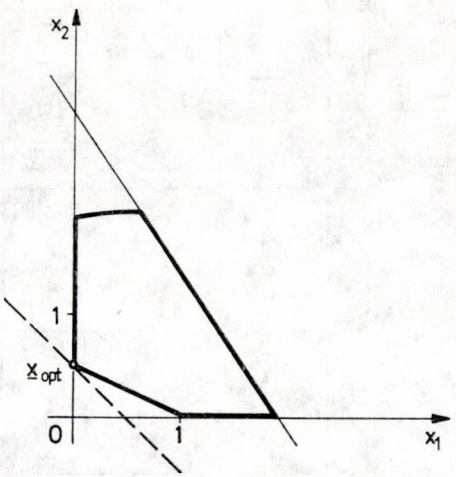
megoldása

$$\mathbf{x}_{\text{opt}} = \begin{pmatrix} 0 \\ 0{,}5 \end{pmatrix}, \quad \text{és} \quad \min (x_1+x_2) = 0{,}5.$$

A sztochasztikus és a determinisztikus feladat megoldását a 3. és a 4. ábra szemlélteti. A két feladat optimális megoldása és az optimum értéke megegyezik.



3. ábra



4. ábra

3. Feladat

$$P(\eta \leqq 4-x_1^2-x_2^2+x_1+x_2) \geqq 0{,}8$$

$$x_1+2x_2 \geqq 1$$

$$-3x_1-2x_2 \geqq -6$$

$$x_1 = 0$$

$$x_2 \geqq 0$$

$$\min (x_1+x_2)$$

3. TÁBLÁZAT

Az iterációk részeredményei

i	$\mathbf{x}_i$	$u(\mathbf{x}_i)$	$G(\mathbf{x}_i)$	$h_1(u(\mathbf{x}_i))$	$(\nabla u(\mathbf{x}_i))'$	$f(\mathbf{x}_i)$	$y_{i\,\mathrm{opt}}$	$\mathbf{x}_i^*$
1	$\begin{pmatrix}0{,}5\\0{,}4\end{pmatrix}$	4,49	0,832 454 8	0,070 828 9	$\begin{pmatrix}0\\0{,}2\end{pmatrix}$	0,9	−0,0389853	$\begin{pmatrix}0\\0{,}861\,014\,6\end{pmatrix}$
2	$\begin{pmatrix}0\\0{,}861\,014\,6\end{pmatrix}$	4,119 668 5	0,802 045 28	0,080 436 8	$\begin{pmatrix}1\\-0{,}722\,029\,2\end{pmatrix}$	0,861 014 6	−0,083 725 1	$\begin{pmatrix}0{,}554\,579\,1\\0{,}222\,710\,4\end{pmatrix}$
3	$\begin{pmatrix}0{,}554\,579\,1\\0{,}222\,710\,4\end{pmatrix}$	4,420 131 6	0,827 009 1	0,072 541 5	$\begin{pmatrix}-0{,}109\,158\,2\\0{,}554\,579\,1\end{pmatrix}$	0,777 289 5	−0,051 633 8	$\begin{pmatrix}0\\0{,}725\,655\,6\end{pmatrix}$
4	$\begin{pmatrix}0\\0{,}725\,655\,6\end{pmatrix}$	4,199 079 5	0,808 890 8	0,078 266	$\begin{pmatrix}1\\-0{,}451\,311\,2\end{pmatrix}$	0,725 655 6	−0,050 471 4	$\begin{pmatrix}0{,}350\,368\,5\\0{,}324\,815\,75\end{pmatrix}$
5	$\begin{pmatrix}0{,}350\,368\,5\\0{,}324\,815\,75\end{pmatrix}$	4,446 920 9	0,829 114 9	0,071 878 2	$\begin{pmatrix}0{,}299\,263\\0{,}350\,368\,5\end{pmatrix}$	0,675 184 25	−0,029 655 1	$\begin{pmatrix}0\\0{,}645\,529\,1\end{pmatrix}$
6	$\begin{pmatrix}0\\0{,}645\,529\,1\end{pmatrix}$	4,228 821 3	0,811 407 7	0,077 469	$\begin{pmatrix}1\\-0{,}291\,058\,2\end{pmatrix}$	0,645 529 1	−0,034 411 04	$\begin{pmatrix}0{,}222\,236\,12\\0{,}388\,881\,94\end{pmatrix}$
7	$\begin{pmatrix}0{,}222\,236\,12\\0{,}388\,881\,94\end{pmatrix}$	4,410 5	0,826 247 46	0,072 781 55	$\begin{pmatrix}0{,}555\,527\,76\\0{,}222\,236\,12\end{pmatrix}$	0,611 118 06	−0,024 652 69	$\begin{pmatrix}0{,}172\,930\,74\\0{,}413\,534\,63\end{pmatrix}$
8	$\begin{pmatrix}0\\0{,}5\end{pmatrix}$	4,25	0,813 184 62	0,076 872 08	$\begin{pmatrix}1\\0\end{pmatrix}$	0,5	0	$\begin{pmatrix}0\\0{,}5\end{pmatrix}$

Az $\eta=\xi_1^2+\xi_2^2$ valószínűségi változó sűrűségfüggvénye (2.3), ahol $\sigma_1=\sqrt{2}$ és $\sigma_2=1$; η várható értéke 3. A $h(y)$ és $H(y)$ függvényeket a $h_1(y)$ és $H_1(y)$ függvények $5\cdot 10^{-4}$ hibával közelítik meg a $0<y\leqq 4{,}5$ intervallumon.

$$h_1(y) = e^{-\frac{y}{4}}(0{,}3535-0{,}0442y+0{,}004\,142y^2-$$
$$-2{,}872\cdot 10^{-4}y^3+1{,}56\cdot 10^{-5}y^4-7\cdot 10^{-7}y^5)$$

$$H_1(y) = 0{,}911\,496\,1-e^{-\frac{y}{4}}(0{,}911\,496\,1-0{,}145\,010\,4y+$$
$$+3{,}969\,7\cdot 10^{-3}y^2-1{,}05\cdot 10^{-3}y^3+6{,}3\cdot 10^{-6}y^4-$$
$$-2{,}8\cdot 10^{-6}y^5).$$

Az $\mathbf{x}_1=\begin{pmatrix}0{,}5\\0{,}4\end{pmatrix}$ vektorra minden feltétel teljesül, és $G(\mathbf{x}_1)>0{,}8$. Az iterációk részeredményeit a 3. táblázat tartalmazza.

Az $y_{8\,\mathrm{opt}}=0$, tehát az optimális megoldás

$$\mathbf{x}_8=\begin{pmatrix}0\\0{,}5\end{pmatrix}, \quad \text{és} \quad \min(x_1+x_2)=0{,}5.$$

A determinisztikus feladat

$$3\leqq 4-x_1^2-x_2^2+x_1+x_2$$
$$x_1+2x_2\geqq 1$$
$$-3x_1-2x_2\geqq -6$$
$$x_1\geqq 0$$
$$x_2\geqq 0$$
$$\min(x_1+x_2)$$

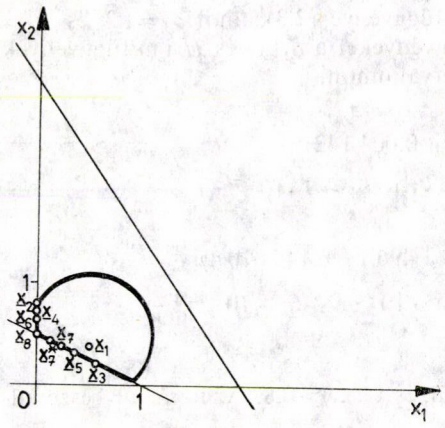
megoldása

$$\mathbf{x}_{\mathrm{opt}}=\begin{pmatrix}0\\0{,}5\end{pmatrix}, \quad \text{és} \quad \min(x_1+x_2)=0{,}5.$$

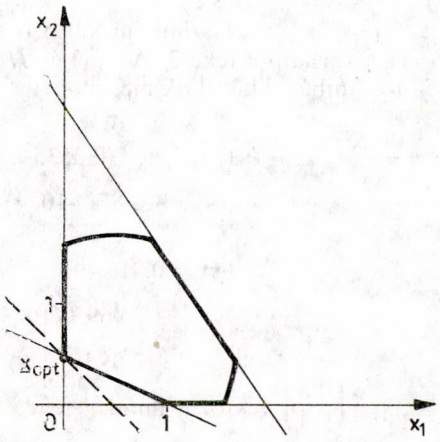
A sztochasztikus és a determinisztikus feladat megoldását az 5. és a 6. ábra szemlélteti. A két feladat optimális megoldása és az optimum értéke megegyezik.

4. Feladat.

$$P(\eta\leqq 4-x_1^2-x_2^2+2x_1+x_2)\geqq 0{,}8$$
$$x_1+2x_2\geqq 1$$
$$-3x_1-2x_2\geqq -6$$
$$x_1\geqq 0$$
$$x_2\geqq 0$$
$$\min(x_1+x_2)$$



5. ábra



6. ábra

Az $\eta=\xi_1^2+\xi_2^2$ valószínűségi változó sűrűségfüggvénye (2.4), ahol $m_1=1$ és $m_2=0$; várható értéke 3. A $h(y)$ és $H(y)$ függvényeket a $h_1(y)$ és a $H_1(y)$ függvények $5\cdot 10^{-4}$ hibával közelítik meg a $0<y\leqq 5{,}25$ intervallumon.

$$h_1(y) = e^{-\frac{y}{2}}(0{,}302\,765\,3+0{,}075\,691\,3y+4{,}730\,7\cdot 10^{-3}y^2+$$
$$+1{,}314\cdot 10^{-4}y^3+2\cdot 10^{-6}y^4)$$

$$H_1(y) = 0{,}998\,336\,9-e^{-\frac{y}{2}}(0{,}998\,336\,9+0{,}196\,403\,1y+$$
$$+0{,}011\,255\,1y^2+2{,}989\cdot 10^{-4}y^3+4{,}5\cdot 10^{-6}y^4).$$

Az $\mathbf{x}_1=\begin{pmatrix}0{,}5\\0{,}5\end{pmatrix}$ vektorra teljesül minden feltétel, és $G(\mathbf{x}_1)>0{,}8$. Az iterációk részeredményeit a 4. táblázat tartalmazza.

Mivel az $\dfrac{f(\mathbf{x}_{10})-f(\mathbf{x}_9)}{f(\mathbf{x}_9)}$ relatív hiba 0%, az optimális megoldás

$$\mathbf{x}_9=\begin{pmatrix}0{,}413\,527\,68\\0{,}293\,236\,16\end{pmatrix}, \quad \text{és} \quad \min(x_1+x_2)=0{,}706\,763\,84.$$

A determinisztikus feladat

$$3 \leqq 4-x_1^2-x_2^2+2x_1+x_2$$
$$x_1+2x_2 \geqq 1$$
$$-3x_1-2x_2 \geqq -6$$
$$x_1 \geqq 0$$
$$x_2 \geqq 0$$
$$\min(x_1+x_2)$$

4. TÁBLÁZAT

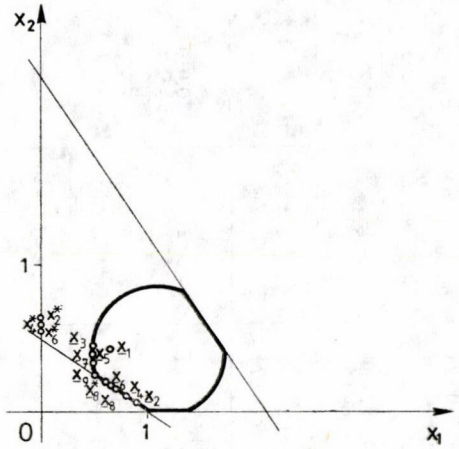
Az iterációk részeredményei

i	$\mathbf{x}_i$	$u(\mathbf{x}_i)$	$G(\mathbf{x}_i)$	$h_1(u(\mathbf{x}_i))$	$(\nabla u(\mathbf{x}_i))'$	$f(\mathbf{x}_i)$	$y_{i\,\mathrm{opt}}$	$\mathbf{x}_i^*$
1	$\begin{pmatrix}0{,}5\\0{,}5\end{pmatrix}$	5	0,809 383 7	0,0670774	$\begin{pmatrix}1\\0\end{pmatrix}$	1	−0,037 845 3	$\begin{pmatrix}0{,}924\,309\,4\\0{,}037\,845\,3\end{pmatrix}$
2	$\begin{pmatrix}0{,}924\,309\,4\\0{,}037\,845\,3\end{pmatrix}$	5,030 683 9	0,811 433	0,066 388 2	$\begin{pmatrix}0{,}151\,381\,2\\0{,}924\,309\,4\end{pmatrix}$	0,962 154 7	−0,055 459 2	$\begin{pmatrix}0\\0{,}906\,695\,4\end{pmatrix}$
3	$\begin{pmatrix}0{,}380\,111\,6\\0{,}549\,390\,7\end{pmatrix}$	4,863 29 9	0,8	0,070 225 2	$\begin{pmatrix}1{,}239\,776\,8\\-0{,}098\,781\,4\end{pmatrix}$	0,929 502 3	−0,038 114 9	$\begin{pmatrix}0{,}782\,774\,6\\0{,}108\,612\,7\end{pmatrix}$
4	$\begin{pmatrix}0{,}782\,774\,6\\0{,}108\,612\,7\end{pmatrix}$	5,049 629	0,812 687	0,065 966 2	$\begin{pmatrix}0{,}434\,450\,8\\0{,}782\,774\,6\end{pmatrix}$	0,891 387 3	−0,029 166 3	$\begin{pmatrix}0\\0{,}862\,221\end{pmatrix}$
5	$\begin{pmatrix}0{,}378\,149\,9\\0{,}498\,161\end{pmatrix}$	4,863 299	0,8	0,070 225 2	$\begin{pmatrix}1{,}243\,700\,2\\0{,}003\,678\end{pmatrix}$	0,876 310 9	−0,027 766 26	$\begin{pmatrix}0{,}697\,089\,28\\0{,}151\,455\,36\end{pmatrix}$
6	$\begin{pmatrix}0{,}697\,089\,28\\0{,}151\,455\,36\end{pmatrix}$	5,036 761 7	0,811 836 2	0,066 252 5	$\begin{pmatrix}0{,}605\,821\,44\\0{,}697\,089\,28\end{pmatrix}$	0,848 544 64	−0,015 342 71	$\begin{pmatrix}0\\0{,}833\,201\,89\end{pmatrix}$
7	$\begin{pmatrix}0{,}379\,293\,6\\0{,}462\,256\,4\end{pmatrix}$	4,863 299	0,8	0,070 225 2	$\begin{pmatrix}1{,}241\,412\,8\\0{,}075\,487\,2\end{pmatrix}$	0,841 55	−0,021 277 7	$\begin{pmatrix}0{,}640\,544\,6\\0{,}179\,727\,7\end{pmatrix}$
8	$\begin{pmatrix}0{,}640\,544\,6\\0{,}179\,727\,7\end{pmatrix}$	5,018 217 4	0,810 603	0,066 667 3	$\begin{pmatrix}0{,}718\,910\,8\\0{,}640\,544\,6\end{pmatrix}$	0,820 272 3	−0,010 067 8	$\begin{pmatrix}0{,}620\,409\\0{,}189\,795\,5\end{pmatrix}$
9	$\begin{pmatrix}0{,}413\,527\,68\\0{,}293\,236\,16\end{pmatrix}$	4,863 299	0,8	0,070 225 2	$\begin{pmatrix}1{,}172\,944\,64\\0{,}413\,527\,68\end{pmatrix}$	0,706 763 84	$-2\cdot10^{-9}$	$\begin{pmatrix}0{,}413\,527\,676\\0{,}293\,236\,162\end{pmatrix}$
10	$\begin{pmatrix}0{,}413\,527\,68\\0{,}293\,236\,16\end{pmatrix}$							

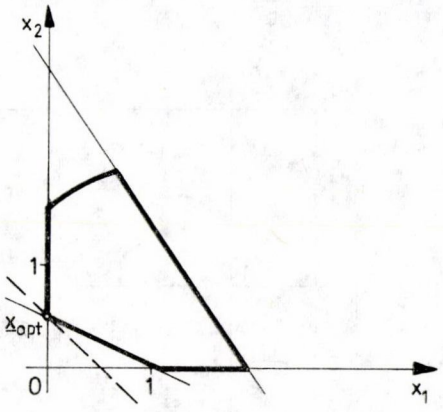
megoldása

$$\mathbf{x}_{opt} = \begin{pmatrix} 0 \\ 0{,}5 \end{pmatrix}, \quad \text{és} \quad \min(x_1+x_2) = 0{,}5.$$

A sztochasztikus és a determinisztikus feladat megoldását a 7. és a 8. ábra szemlélteti. A két feladat optimális megoldása és az optimum értéke különböző.



7. ábra



8. ábra

5. Feladat.

$$P(\eta \leqq 3-x_1^2-x_2^2+x_1+x_2) \geqq 0{,}8$$
$$x_1+2x_2 \geqq 1$$
$$-3x_1-2x_2 \geqq -6$$
$$x_1 \geqq 0$$
$$x_2 \geqq 0$$
$$\min(x_1+x_2)$$

Az $\eta=\xi_1^2+\xi_2^2$ valószínőségi változó eloszlásfüggvénye (2.5), várható értéke 2. A $h(y)$ és $H(y)$ függvényeket a $h_1(y)$ és a $H_1(y)$ függvények $5\cdot 10^{-4}$ hibával közelítik meg a $0<y\leqq 3{,}5$ intervallumon.

$$\begin{aligned} h_1(y) = {} & 0{,}577\,35-0{,}384\,9y+0{,}144\,336y^2- \\ & -0{,}039\,202\,4y^3+8{,}427\cdot 10^{-3}y^4-1{,}499\,58\cdot 10^{-3}y^5+ \\ & +2{,}412\,9\cdot 10^{-4}y^6-3{,}002\,16\cdot 10^{-5}y^7+ \\ & +3{,}508\,29\cdot 10^{-6}y^8-3{,}672\,3\cdot 10^{-7}y^9+ \\ & +3{,}482\,05\cdot 10^{-8}y^{10}-3{,}451\cdot 10^{-9}y^{11}+ \\ & +2{,}406\,8\cdot 10^{-10}y^{12}, \end{aligned}$$

$$H_1(y) = 0{,}57735y - 0{,}19245y^2 + 0{,}048112y^3 -$$
$$- 9{,}8006 \cdot 10^{-3} y^4 + 1{,}6854 \cdot 10^{-3} y^5 - 2{,}4993 \cdot 10^{-4} y^6 +$$
$$+ 3{,}2447 \cdot 10^{-5} y^7 - 3{,}7527 \cdot 10^{-6} y^8 + 3{,}8981 \cdot 10^{-7} y^9 -$$
$$- 3{,}6723 \cdot 10^{-8} y^{10} + 3{,}1655 \cdot 10^{-9} y^{11} - 2{,}8759 \cdot 10^{-10} y^{12} +$$
$$+ 1{,}8514 \cdot 10^{-11} y^{13}.$$

Az $\mathbf{x}_1 = \begin{pmatrix} 0{,}2 \\ 0{,}4 \end{pmatrix}$ vektorra minden feltétel teljesül, és $G(\mathbf{x}_1) > 0{,}8$. Az iterációk részeredményeit az 5. táblázat tartalmazza.

Az $y_{opt} = 0$, tehát a feladat optimális megoldása

$$\mathbf{x}_2 = \begin{pmatrix} 0 \\ 0{,}5 \end{pmatrix}, \quad \text{és} \quad \min(x_1 + x_2) = 0{,}5.$$

A determinisztikus feladat

$$2 \leqq 3 - x_1^2 - x_2^2 + x_1 + x_2$$
$$x_1 + 2x_2 \geqq 1$$
$$-3x_1 - 2x_2 \geqq -6$$
$$x_1 \geqq 0$$
$$x_2 \geqq 0$$
$$\min(x_1 + x_2)$$

megoldása

$$\mathbf{x}_{opt} = \begin{pmatrix} 0 \\ 0{,}5 \end{pmatrix}, \quad \text{és} \quad \min(x_1 + x_2) = 0{,}5.$$

A sztochasztikus és a determinisztikus feladat megoldását a 9. és a 10. ábra szemlélteti. A két feladat optimális megoldása és az optimum értéke megegyezik.

6. Feladat.

$$P(\eta \leqq 4 - x_1^2 - x_2^2 + x_1 + x_2) \geqq 0{,}8$$
$$x_1 + 2x_2 \geqq 1$$
$$-3x_1 - 2x_2 \geqq -6$$
$$x_1 \geqq 0$$
$$x_2 \geqq 0$$
$$\min(x_1 + x_2)$$

Az $\eta = \xi_1^2 + \xi_2$ valószínűségi változó sűrűségfüggvénye (2.6), várható értéke 1.

5. TÁBLÁZAT

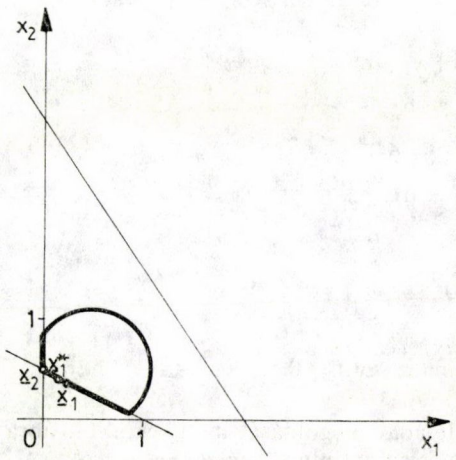
Az iterációk részeredményei

i	$\mathbf{x}_i$	$u(\mathbf{x}_i)$	$G(\mathbf{x}_i)$	$h_1(u(\mathbf{x}_i))$	$(\nabla u(\mathbf{x}_i))'$	$f(\mathbf{x}_i)$	$y_{i\,\mathrm{opt}}$	$\mathbf{x}_i^*$
1	$\begin{pmatrix}0{,}2\\0{,}4\end{pmatrix}$	3,4	0,820 428 095	0,101 952 496	$\begin{pmatrix}0{,}6\\0{,}2\end{pmatrix}$	0,6	−0,018 538 09	$\begin{pmatrix}0{,}162\,923\,822\\0{,}418\,538\,029\end{pmatrix}$
2	$\begin{pmatrix}0\\0{,}5\end{pmatrix}$	3,25	0,807 934 648	0,103 279 844	$\begin{pmatrix}1\\0\end{pmatrix}$	0,5	0	$\begin{pmatrix}0\\0{,}5\end{pmatrix}$

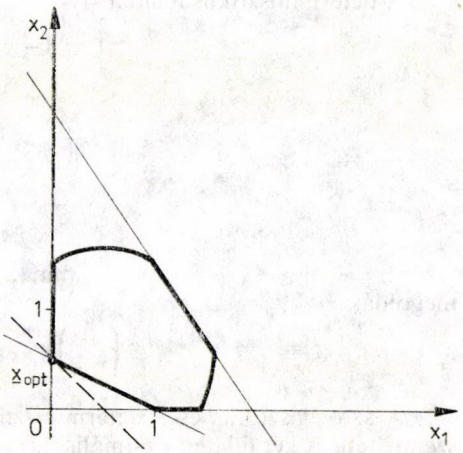
6. TÁBLÁZAT

Az iterációk részeredményei

i	$\mathbf{x}_i$	$u(\mathbf{x}_i)$	$G(\mathbf{x}_i)$	$h_1(u(\mathbf{x}_i))$	$(\nabla u(\mathbf{x}_i))'$	$f(\mathbf{x}_i)$	$y_{i\,\mathrm{opt}}$	$\mathbf{x}_i^*$
1	$\begin{pmatrix}0{,}2\\0{,}4\end{pmatrix}$	4,4	0,976 845 67	0,009 766 17	$\begin{pmatrix}0{,}6\\0{,}2\end{pmatrix}$	0,6	−0,1	$\begin{pmatrix}0\\0{,}5\end{pmatrix}$
2	$\begin{pmatrix}0\\0{,}5\end{pmatrix}$	4,25	0,975 100 55	0,013 739 08	$\begin{pmatrix}1\\0\end{pmatrix}$	0,5	0	$\begin{pmatrix}0\\0{,}5\end{pmatrix}$



9. ábra



10. ábra

A $h(y)$ és $H(y)$ függvényeket a $h_1(y)$ és a $H_1(y)$ függvények $5 \cdot 10^{-3}$ hibával közelítik meg az $|y| \leqq 4{,}5$ intervallumon.

$$h_1(y) = e^{-\frac{y^2}{2}}(0{,}276\,59 + 0{,}103y + 0{,}043\,397y^2 +$$
$$+ 0{,}018\,512y^3 + 6{,}743\,4 \cdot 10^{-3}y^4 + 2{,}518\,3 \cdot 10^{-3}y^5 +$$
$$+ 8{,}907\,4 \cdot 10^{-4}y^6 + 1{,}488\,4 \cdot 10^{-4}y^7 + 9{,}4 \cdot 10^{-5}y^8 +$$
$$+ 6{,}856 \cdot 10^{-5}y^9 + 2{,}926 \cdot 10^{-5}y^{10}),$$

$$H_1(y) = 0{,}391\,099 \int_{-\infty}^{y} e^{-\frac{z^2}{2}}\,dz - e^{-\frac{y^2}{2}}(0{,}193\,642\,528 + 0{,}114\,509y + 0{,}045\,321\,26y^2 +$$
$$+ 0{,}023\,704y^3 + 6{,}702\,31 \cdot 10^{-3}y^4 + 3{,}392\,12 \cdot 10^{-3}y^5 +$$
$$+ 6{,}973\,36 \cdot 10^{-4}y^6 + 3{,}573\,4 \cdot 10^{-4}y^7 + 6{,}856\,2 \cdot 10^{-5}y^8 +$$
$$+ 2{,}926 \cdot 10^{-5}y^9).$$

Az $\mathbf{x}_1 = \begin{pmatrix} 0{,}2 \\ 0{,}4 \end{pmatrix}$ vektorra minden feltétel teljesül, és $G(\mathbf{x}_1) > 0{,}8$. Az iterációk részeredményeit a 6. táblázat tartalmazza.

Az $y_{\text{opt}} = 0$, tehát a feladat optimális megoldása

$$\mathbf{x}_2 = \begin{pmatrix} 0 \\ 0{,}5 \end{pmatrix}, \quad \text{és} \quad \min(x_1 + x_2) = 0{,}5.$$

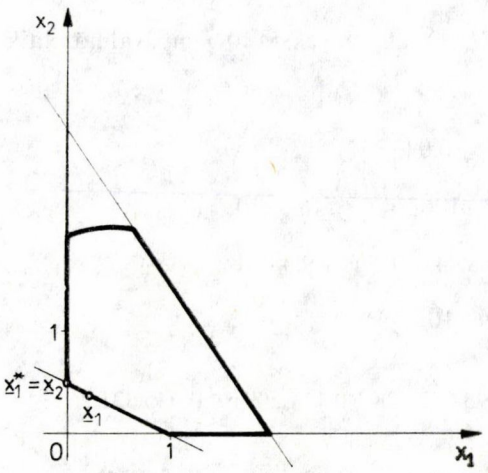
A determinisztikus feladat

$$1 \leqq 4-x_1^2-x_2^2+x_1+x_2$$
$$x_1+2x_2 \geqq 1$$
$$-3x_1-2x_2 \geqq -6$$
$$x_1 \geqq 0$$
$$x_2 \geqq 0$$
$$\min (x_1+x_2)$$

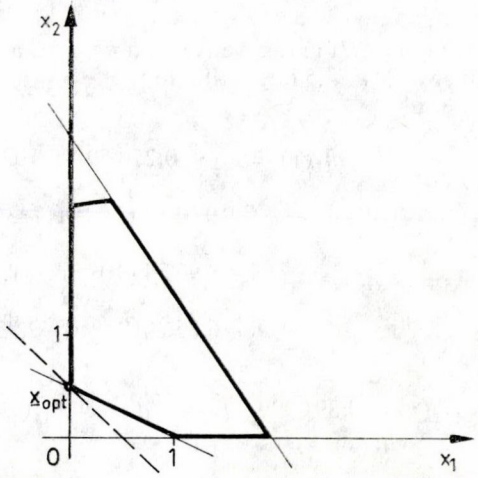
megoldása

$$\mathbf{x}_{\text{opt}} = \begin{pmatrix} 0 \\ 0{,}5 \end{pmatrix}, \quad \text{és} \quad \min (x_1+x_2) = 0{,}5.$$

A sztochasztikus és a determinisztikus feladat megoldását a 11. és a 12. ábra szemlélteti. A két feladat optimális megoldása és az optimum értéke megegyezik.



11. ábra



12. ábra

IRODALOM

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

[2] PRÉKOPA, A., "A class of stochastic programming decision problems ", *Mathematische Operationsforschung und Statistik* **3** (1972) 349—354.

[3] PRÉKOPA, A., "Contributions to the theory of stochastic programming", *Mathematical Programming* **4** (1973) 202—221.

[4] PRÉKOPA, A., GANCZER, S., DEÁK, I. és PATYI, K., „A STABIL sztochasztikus programozási modell és annak kísérleti alkalmazása a magyar villamosenergiaiparra", *Alkalmazott Matematikai Lapok* **1** (1975).

[5] PRÉKOPA, A., „A logkonkáv mértékek alaptételének új bizonyítása", *Alkalmazott Matematikai Lapok* **1** (1976).

[6] MAYER, J., „A STABIL sztochasztikus programozási modellről", *Alkalmazott Matematikai Lapok* 2 (1976) 171—187.
[7] SZÁNTAI, T., „A Prékopa-féle STABIL matematikai programozási modell numerikus megoldásáról", *Alkalmazott Matematikai Lapok* 2 (1976) 93—101.
[8] HARWIDE, D. A., "On the distribution of linear combinations of non-central chi-squares", *The Annals of Mathematical Statistics* 42 (1971) 809—811.
[9] ZOUTENDIJK, G., *Methods of Feasible Directions* (Elsevier Publishing Company, Amsterdam—London—New York—Princeton, 1960).
[10] HADLEY, G., *Nichtlineare und dynamische Programmierung* (Berlin, 1969).

(Beérkezett: 1980. április 24).

STUBNYA GUSZTÁVNÉ
BME VILLAMOSMÉRNÖKI KAR MATEMATIKA TANSZÉK
1111 BUDAPEST, STOCZEK U. H ÉP. III. E.

NUMERICAL EXAMPLES FOR PROBABILISTIC CONSTRAINED STOCHASTIC PROGRAMMING PROBLEMS

E. STUBNYA

In this paper we give the solution of some probabilistic constrained stochastic programming problems. These problems are more general than those in paper [4] as we have quadratic function in the probabilistic constraint. Our first aim is to give examples on the solution of such stochastic programming problems. The numerical example are of small size so we can follow the iterations of the solution procedure.

Alkalmazott Matematikai Lapok 6 (1980) 257—264

EGY PROBLÉMÁRÓL, AMELY INTERVALLUMOK EGYENLETES FELOSZTÁSAIVAL KAPCSOLATOS

ÁDÁM ANDRÁS

Budapest

Bontsuk fel a $[0, k]$ intervallumot k, illetve $n(>k)$ darab egyenlő hosszúságú szakaszra, jelöljük e felosztásokat F_1-gyel, illetve F_2-vel. Válasszunk egy r valós számot $(0<r\leqq k/n)$. Képletet adunk arra (5.1. tétel), hogy a

$$[0, r], [1, 1+r], [2, 2+r], \ldots, [k-1, k-1+r]$$

szakaszok között hány olyan van, amelyet F_2 egyben hagy.

1. Jelölések

A cikkben az a, c, h, i, j, k, n, m betűk egész számokat, az r, s, t, x, y betűk valós számokat jelentenek. Ha $x=ay$ igaz valamilyen (egész) a-val, akkor x-et y többszörösének és y-t x osztójának mondjuk. Ha x és y racionális arányban vannak egymással, akkor beszélhetünk e két szám legnagyobb közös osztójáról, (x, y)-ról.[1]

A lefelé kerekítés jelölésére az $[x]$, a felfelé kerekítésére az $[x]$ szimbólumot használjuk. (Tehát $[x]$ és $[x]$ azok az egész számok, amelyekre $x-1<[x]\leqq x$, illetve $x\leqq[x]<x+1$ igaz.)

A zárt intervallumot a szokásos $[x, y]$ jellel jelöljük, a nyílt intervallum jelölésére (a legnagyobb közös osztótól való megkülönböztetés céljából) $\langle x, y\rangle$ szolgál.

Az $x-[x]$ különbséget (x törtrészét) $\{x\}$-szel jelöljük.

2. A probléma

Tekintsünk két n, k egész számot, amelyekre $0<k<n$ igaz. Legyen továbbá r olyan szám, hogy $0<r<1$. Induljunk ki a számegyenes $[0, k]$ intervallumának k részre való egyenletes felosztásából; ennek az F_1 felosztásnak nyilván $1, 2, \ldots, k-1$ lesznek a határpontjai. Osszuk fel ezután ugyanezt a $[0, k]$ intervallumot n egyenlő részre; ez az F_2 felosztás a

$$k/n, 2k/n, 3k/n, \ldots, (n-1)k/n$$

határpontokat szolgáltatja.

Válasszunk egy h számot $(0\leqq h<k)$. A $[h, h+1]$ intervallumot (az F_1 felosztás szerinti szakaszok egyikét) *szabályos*nak nevezzük, ha az abban foglalt $[h, h+r]$ inter-

[1] Többnyire egész számoknak fogjuk a legnagyobb közös osztóját képezni. x és y legkisebb közös többszörösére nem vezetünk be külön jelölést, mivel ez az $xy/(x, y)$ hányadossal egyenlő.

vallumot F_2 egyben hagyja (azaz ha $[h, h+r]$ az F_2 felosztás határpontjainak egyikét sem tartalmazza belső pontjaként).

A probléma abban áll, hogy *hány szabályos szakasz létezik adott n, k, r számok esetén.* Jelöljük a szabályos szakaszok számát $f(n, k, r)$-rel. Világos, hogy $r>k/n$ esetén $f(n, k, r)=0$, tehát a probléma csak az $r \leqq k/n\ (<1)$ feltétel teljesülése mellett érdekes. Az is nyilvánvaló, hogy $f(n, k, r) \leqq k$; felvethető a kérdés, mikor igaz az egyenlőség ebben a becslésben.

*

A leírt problémára egy gyakorlati kérdés vezetett, amelynek lehetséges matematikai vonatkozásaira CSIKI RÓBERT és SZERDAHELYI JUDIT építészmérnökök hívták fel figyelmemet (v.ö. [1]). A gyakorlati kérdés a következőképpen vázolható. Valamely utca-szakasz (az utca egyik oldala) eléggé szabályosan van beépítve abban az értelemben, hogy jó közelítéssel igaznak vehetőek az alábbiak:

(a) a telkek szélessége (az utcára eső határvonal hossza) egyenlő egymással,

(b) a telkeken házak vannak, amelyeknek az utcával párhuzamos hosszmérete szintén közös, és

(c) a ház mindegyik telken ugyanúgy helyezkedik el, éspedig mindig a telek bal szélén kezdődik.

Az eredetileg adott k telek helyett $n(>k)$ részre akarjuk parcellázni az utcavonalat, úgy hogy újra igaz legyen az (a) tulajdonság, és hogy a házak egyikét se vágja ketté az új határvonalak valamelyike.

3. Előkészületek

Támaszkodni fogunk arra a tényre, hogy a

$$\{k/n\},\ \{2k/n\},\ \ldots,\ \{(n-1)k/n\} \tag{3.1}$$

számsorozatban levő számok mindegyike $1/n$ többszöröse (ez az

$$\left\{\frac{mk}{n}\right\} = \frac{mk}{n} - \left\lfloor\frac{mk}{n}\right\rfloor$$

egyenlőség révén látható).

3.1. SEGÉDTÉTEL. Ha n, k relatív prímek, akkor a

$$\{0k/n\}(=0),\ \{k/n\},\ \{2k/n\},\ \ldots,\ \{(n-1)k/n\} \tag{3.2}$$

számsorozatban levő számok páronként különbözőek.

Bizonyítás. Tegyük fel, hogy a (3.2) sorozat tagjai között van két egybeeső. Ez azt jelenti, hogy létezik két i, j szám úgy, hogy

$$0 \leqq i < j < n \tag{3.3}$$

és $\{ik/n\}=\{jk/n\}$. Ekkor a

$$\frac{jk}{n}-\frac{ik}{n}$$

különbség egy *egész* számmal egyenlő, mondjuk, a-val.

Ez az egyenlőség

$$an=(j-i)k \tag{3.4}$$

alakra hozható. Mivel (3.4) igaz és n, k relatív prímek, $j-i$ többszöröse n-nek, de ez (3.3) miatt lehetetlen.

3.2. SEGÉDTÉTEL. Ha n, k relatív prímek, akkor a (3.1) sorozatban levő számok sorrendtől eltekintve megegyeznek az

$$1/n, 2/n, 3/n, \ldots, (n-1)/n \tag{3.5}$$

számokkal.

Bizonyítás. A 3.1. segédtétel értelmében a (3.1) sorozat $n-1$ darab különböző számból áll; ezek mind a $\langle 0, 1\rangle$ nyílt intervallumban vannak és mind többszörösei $1/n$-nek. Ennélfogva (3.1)-ben éppen a (3.5) alatt felsorolt számok vannak valamilyen sorrendben.

4. Megoldás relatív prím n és k esetén

4.1. TÉTEL. Ha n és k relatív prímek, akkor

$$f(n, k, r) = k-[rn]+1.$$

Bizonyítás. Fusson végig i azokon az egész számokon, amelyekre

$$[rn] \leqq i \leqq k. \tag{4.1}$$

Ilyen szám nyilván $k-[rn]+1$ darab van.

A további bizonyítás három részre tagolódik. Az első részben minden i-hez hozzárendelünk egy $[h, h+1]$ intervallumot, és kimutatjuk, hogy az szabályos. A második részben azt igazoljuk, hogy különböző i-khez különböző szakaszok tartoznak. A harmadik részben bebizonyítjuk, hogy nincs más szabályos intervallum, mint azok, amelyeket az I. részben előállítottunk.

I. Tegyen eleget i a (4.1) feltételnek. Tekintsük az F_2 felosztásnak azt az mk/n határpontját, amelyre

$$\left\{\frac{mk}{n}\right\}=\frac{i}{n}.$$

(Ez a határpont a 3.2 segédtétel értelmében létezik és egyértelmű.) Jelöljük az $[mk/n]$ számot h-val.

A $[h, h+1]$ intervallum szabályos volta következik az

$$\frac{(m-1)k}{n} \leqq h, \quad h+r \leqq \frac{mk}{n} \tag{4.2}$$

egyenlőtlenségekből; az tehát a célunk, hogy ezt a két egyenlőtlenséget belássuk. A (4.2)-beli első formula azért érvényes, mert

$$\frac{(m-1)k}{n} = \frac{mk}{n} - \frac{k}{n} = \left\lfloor \frac{mk}{n} \right\rfloor + \left\{ \frac{mk}{n} \right\} - \frac{k}{n} =$$

$$= h + \frac{i}{n} - \frac{k}{n} = h - \frac{k-i}{n} \leqq h;$$

a második azért igaz, mert

$$h + r = h + \frac{rn}{n} \leqq h + \frac{[rn]}{n} \leqq h + \frac{i}{n} = \left\lfloor \frac{mk}{n} \right\rfloor + \left\{ \frac{mk}{n} \right\} = \frac{mk}{n}.$$

II. Tekintsünk két i_1, i_2 számot, amelyekre

$$[rn] \leqq i_1 < i_2 \leqq k \tag{4.3}$$

teljesül, és nézzük a hozzájuk rendelt $[h_1, h_1+1]$, $[h_2, h_2+1]$ intervallumokat.

Ki akarjuk mutatni, hogy a $h_1 = h_2$ feltevés ellentmondásra vezet. Valóban, ekkor

$$\frac{i_2 - i_1}{n} = \frac{i_2}{n} - \frac{i_1}{n} = \left\{ \frac{m_2 k}{n} \right\} - \left\{ \frac{m_1 k}{n} \right\} =$$

$$= \left(\frac{m_2 k}{n} - h_2 \right) - \left(\frac{m_1 k}{n} - h_1 \right) = (m_2 - m_1) \frac{k}{n}$$

(a feltevést a levezetés utolsó lépésében használtuk ki), tehát

$$i_2 - i_1 = (m_2 - m_1) k,$$

és így vagy $i_1 = i_2$, vagy $i_2 - i_1 \geqq k$ igaz. A (4.3) feltétellel mindenképpen ellentmondásba kerülünk.

III. Tekintsünk egy $[h, h+1]$ szabályos intervallumot. Meg akarjuk határozni azt az i számot, amelyből kiindulva a bizonyítás első része éppen a $[h, h+1]$ szakaszt szolgáltatja.

Legyen m a legkisebb olyan egész szám, amelyre

$$\frac{mk}{n} > h \tag{4.4}$$

teljesül.[2] Az intervallum szabályos voltából a (4.2) egyenlőtlenségek adódnak. Jelöljük az $\left\{ \frac{mk}{n} \right\} n$ számot i-vel.

Legközelebbi célunk azt belátni, hogy i eleget tesz (4.1)-nek. A (4.2) képletek szerint

$$h + r \leqq \frac{mk}{n} = \frac{(m-1)k}{n} + \frac{k}{n} < h + 1,$$

[2] Éspedig $m = [hn/k] + 1$, de m-nek erre az előállítására nem lesz szükségünk.

ebből $\{mk/n\} \leqq r$ következik, tehát

$$rn \leqq \left\{\frac{mk}{n}\right\} n = i,$$

ennélfogva $[rn] \leqq i$ (mivel i egész szám). Másrészt a (4.4)-ből adódó $h \leqq [mk/n]$ egyenlőtlenség és (4.2) folytán

$$\left\{\frac{mk}{n}\right\} = \frac{mk}{n} - \left[\frac{mk}{n}\right] \leqq \frac{(m-1)k}{n} + \frac{k}{n} - h \leqq \frac{k}{n},$$

és így

$$i = \left\{\frac{mk}{n}\right\} n \leqq k.$$

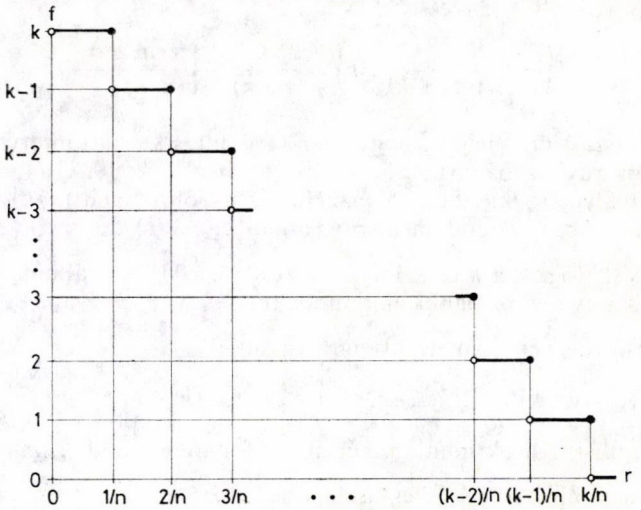
Tehát i teljesíti (4.1)-et.

Az m, n, k, h számokra igaz a

$$0 \leqq \frac{mk}{n} - h = \frac{k}{n} - \left(h - \frac{(m-1)k}{n}\right) \leqq \frac{k}{n} < 1$$

levezetés, tehát $h = [mk/n]$. Meggondolásaink azt mutatják, hogy a jelen részben szereplő $h \to i$ hozzárendelés inverze a bizonyítás első részében tekintett $i \to h$ hozzárendelésnek.

Megjegyzés. Rögzítsük az n és k számokat, futtassuk r-et végig a $\langle 0, k/n]$ intervallumon. A 4.1 tétel alapján $f(n, k, r)$ viselkedése a következőképpen írható le (l. 1. ábra):



1. ábra

(1) f lépcsősfüggvény, amelynek ugrási pontjai

$$1/n,\ 2/n,\ 3/n,\ \ldots,\ (k-1)/n;$$

(2) f értéke az ugrási pontok mindegyikében bal-folytonos módon eggyel csökken;

(3) f legnagyobb értéke k és legkisebb értéke 1.

4.1. KÖVETKEZMÉNY. Relatív prím n, k számok esetén $f(n, k, r)=k$ akkor és csak akkor igaz, ha $0<r\leqq 1/n$.

5. Megoldás az általános esetben

5.1. TÉTEL. Ha n és k tetszőleges egész számok, amelyekre $0<k<n$ teljesül, akkor

$$f(n, k, r) = k-(n, k)\left(\left[\frac{rn}{(n, k)}\right]-1\right).$$

Bizonyítás. Osszuk fel a $[0, k]$ intervallumot (n, k) egyenlő részre. Ennek az F_3 felosztásnak

$$k/(n, k),\ 2k/(n, k),\ 3k/(n, k),\ \ldots,\ k\left(1-1/(n, k)\right) \tag{5.1}$$

lesznek a határpontjai. Az (5.1)-beli számok közös határpontjai F_1-nek és F_2-nek; továbbá (minthogy 1 és k/n legkisebb közös többszöröse éppen $k/(n, k)$) e két felosztásnak nincs több közös határpontja, mint az (5.1)-ben felsoroltak.

Alkalmazzuk a 4.1. tételt az F_3 felbontás első szakaszára úgy, hogy n és k szerepét $n/(n, k)$, illetve $k/(n, k)$ játssza. Azt kapjuk, hogy F_1-nek a $[0, k/(n, k)]$ intervallumon belül

$$f\left(\frac{n}{(n, k)},\ \frac{k}{(n, k)},\ r\right) = \frac{k}{(n, k)}-\left[\frac{rn}{(n, k)}\right]+1 \tag{5.2}$$

szabályos szakasza van. Világos, hogy az F_3 felbontás szerinti intervallumok mindegyikébe ugyanennyi szabályos szakasz esik, tehát a teljes $[0, k]$ intervallum (n, k)-szor ennyi szabályos szakaszt tartalmaz. Ha (5.2) jobb oldalát (n, k)-val szorozzuk, akkor éppen a tételben szereplő mennyiség adódik.

Megjegyzés. Rögzített n és k mellett r-et a $\langle 0, k/n]$ intervallumon végigfuttatva az $f(n, k, r)$ mennyiség az alábbi módon alakul:

(1) f lépcsősfüggvény, amelynek ugrási pontjai

$$(n, k)/n,\ 2(n, k)/n,\ 3(n, k)/n,\ \ldots,\ \big(k-(n, k)\big)/n;$$

(2) f az ugrási pontok mindegyikében bal-folytonos módon (n, k)-val csökken;

(3) f legnagyobb értéke k és legkisebb értéke (n, k).

5.1. KÖVETKEZMÉNY. $f(n, k, r)=k$ akkor és csak akkor igaz, ha $0<r\leqq(n, k)/n$.

6. Változatok

A 2—5. pontokban tárgyalt alapproblémának egy általánosítását[3] képezi a következő kérdés. Legyen t tetszőleges pozitív valós szám. Osszuk fel a $[0, t]$ intervallumot k, illetve n egyenlő szakaszra $(0<k<n)$, jelöljük e két felosztást F_1-gyel és F_2-vel. A felosztások szakaszhosszaira vezessük be az $x=t/k$ és $y=t/n$ jelöléseket. Válasszunk egy s számot $(0<s<y)$; nevezzük szabályosnak a $[hx, hx+x]$ intervallumot, ha annak $[hx, hx+s]$ részintervallumát F_2 egyben hagyja $(0\leqq h<k)$.

Mennyi ekkor a szabályos intervallumok $g(t, n, k, s)$ száma? Azáltal, hogy a jelen problémánk hosszúságegysége helyett x-et választjuk egységül, visszavezetjük a kérdést az 5.1. tételben megválaszolt problémára, éspedig azt kapjuk, hogy

$$g(t, n, k, s) = f\left(n, k, \frac{s}{x}\right) =$$

$$= k-(n, k)\left(\left[\frac{sn}{x(n, k)}\right]-1\right) = k-(n, k)\left(\left[\frac{snk}{t(n, k)}\right]-1\right).$$

Felvethetjük úgy is a problémát, hogy a keresett $g(t, n, k, s)$ mennyiséget nem t, n, k, s függvénye gyanánt, hanem c, x, y, s függvényeként akarjuk előállítani, ahol c-vel az (n, k) legnagyobb közös osztót jelöljük. Felhasználva x és y definícióját, valamint a

$$\frac{t}{c} = \frac{xy}{(x, y)}$$

egyenlőséget — amely azért igaz, mert mindkét oldalon az F_1, F_2 felosztások legkisebb közös (pozitív) határpontja áll — azt kapjuk, hogy

$$t = \frac{cxy}{(x, y)}, \quad k = \frac{cy}{(x, y)}, \quad n = \frac{cx}{(x, y)},$$

tehát a keresett $g^*(c, x, y, s)$ függvényre

$$g^*(c, x, y, s) = g\left(\frac{cxy}{(x, y)}, \frac{cx}{(x, y)}, \frac{cy}{(x, y)}, s\right) = \frac{cy}{(x, y)} - c\left(\left[\frac{s}{(x, y)}\right]-1\right)$$

igaz. Ebből az is látható, hogy a függvényérték éppen az $s\leqq(x, y)$ esetben lesz a lehető legnagyobb.

A nyert képleteknek konkrét esetben való tanulmányozására a $t=120$, $n=15$, $k=12$ értékek adnak egy kényelmes lehetőséget (s-et a $\langle 0,8]$ intervallumból választhatjuk).

[3] Inkább látszólagos, mint valódi általánosításról van szó. Csupán annyit módosítunk ugyanis a problémán, hogy elejtjük a hosszúságegység speciális megválasztását (azt, hogy az F_1 felosztás szakaszhossza számít egységnek).

7. Egy nyílt kérdés

Az eddigiekben mindig úgy fogalmaztuk meg a problémát, hogy F_1 szakaszhossza eleve racionális arányban volt F_2 szakaszhosszával. Az eredeti problémánál nehezebbnek látszó kérdéshez jutunk akkor, ha a szakaszhosszak aránya tetszőleges lehet. Az egyszerűség kedvéért újra megállapodunk abban, hogy — akárcsak a 4. és 5. pontokban — F_1 szakaszhosszát vesszük hosszúságegységnek.

Tekintsük az y és r számokat, amelyekre $0<r<y<1$ igaz. Jelöljük $\varphi(k, y, r)$-rel a

$$[0, r],\ [1, 1+r],\ [2, 2+r],\ \ldots,\ [k-1, k-1+r]$$

szakaszok közül azoknak a számát, amelyeknek y egyetlen többszörösét sem tartalmazzák belső pontként. Értelmezzük a $\psi(y, r)$ függvényt

$$\psi(y, r) = \lim_{k\to\infty} \frac{\varphi(k, y, r)}{k}$$

által.

Mit mondhatunk $\psi(y, r)$ létezéséről és tulajdonságairól?

Tekintsük azt a speciális esetet, amikor y racionális (és y redukált előállítása m/a), valamint k többszöröse y-nak. Ekkor, a k/y hányadost n-nel jelölve, azt kapjuk, hogy az (n, k) legnagyobb közös osztó k/m-mel egyenlő, és az 5.1. tétel révén

$$\varphi(k, y, r) = f(n, k, r) = k - \frac{k}{m}([ar]-1).$$

Ennélfogva racionális y mellett

$$\psi(y, r) = 1 - \frac{[ar]-1}{m}$$

érvényes, ha $\psi(y, r)$ egyáltalán létezik.

IRODALOM

[1] CSIKI, R. és SZERDAHELYI, J., „Hagyományos családiházas területek átépítése korszerű csoportházakkal", *Városépítés*, 1978. május, 8—9.

(Beérkezett: 1980. július 30.)

ÁDÁM ANDRÁS
MTA MATEMATIKAI KUTATÓ INTÉZET
1053 BUDAPEST, REÁLTANODA U. 13—15.

ÜBER EIN PROBLEM, DAS GLEICHMÄSSIGE ZERTEILUNGEN VON INTERVALLEN BETRIFFT

A. ÁDÁM

Zerteilen wir das Intervall $[0, k]$ in k bzw. $n(>k)$ Strecken von gleicher Länge; bezeichnen wir diese Zerteilungen durch F_1 bzw. F_2. Sei eine reelle Zahl r gewählt $(0<r\leqq k/n)$. Eine Formel wird (Satz 5.1) dafür gegeben, wie viele Strecken unter

$$[0, r],\ [1, 1+r],\ [2, 2+r],\ \ldots,\ [k-1, k-1+r]$$

existieren, die durch F_2 in einem gelassen werden.

Alkalmazott Matematikai Lapok 6 (1980) 265—271

A HELLINGER-TÁVOLSÁG EGY ALKALMAZÁSÁRÓL

JUHÁSZ FERENC

Budapest

A dolgozat a *Hellinger-* (arccos) *távolságot* teszi vizsgálat tárgyává abból a szempontból, hogy mennyire fejezi ki a populációk közötti különbségeket. A *Hellinger-távolságot* géngyakoriság becslésére használva a maximum likelihood becsléshez *(Bernstein-módszer)* nagyon közeli eredményeket kapunk. (Ezenkívül fel szeretnénk hívni a figyelmet arra, hogy paleoszerológiai vizsgálatok esetén a MOURANT et al. könyvben leírt *Dobson/Ikin-módszer* — melyet ott *Fisher-módszer* néven említenek — használata nem indokolt.)

1.

A régészeti kutatások egyik fontos segédeszköze az antropológiai vizsgálat. Ezen belül az *ABO* vércsoport tulajdonsággal fogunk foglalkozni, amely, mint ismeretes, meghatározható a különböző korokból származó csontmaradványokból.

Az *ABO* vércsoport tulajdonságot egy három allélos gén határozza meg, mely gének közül kettő (*A*, *B*) domináns, a harmadik (*O*) recesszív. Az *AA*, *AO*, *BB*, *BO*, *OO* és *AB* genotípusokból a következő módon alakulnak a fenotípusok: az *AA* és *AO* genotípusok alkotják az *A* fenotípust, a *BB* és *BO* a *B* fenotípust, az *OO* az *O* fenotípust, míg az *AB* fenotípust is jelöl.

A genetika egyik fontos absztrakciója a *Hardy—Weinberg (H—W)-féle equilibrium* [2]. Egy populáció akkor alkot *H—W-féle equilibriumot,* ha

1) zárt, azaz nincsen se ki-, se bevándorlás,
2) érvényesül a véletlen párválasztás elve, azaz nincsenek „kasztok" a populáción belül,
3) egyforma esély utód létrehozására,
4) nincs mutáció.

Az ilyen populációra vonatkozik a *Hardy—Weinberg-féle törvény,* amely szerint a populáció génállománya az idő múlásával nagy valószínűséggel nem változik.

2.

Populációk (temetők, korok) géngyakoriságainak meghatározására számos módszer ismeretes [2], [4]. Ezek közül a *Bernstein-módszer* [1] egyszerűségével tűnik ki. Jelölje *A*, *B*, *O* és *AB* a megfigyelt fenotípusok számát. Az *A*, *B* és *O* gének *p*, *q* és *r*

gyakoriságait *Bernstein módszerével* a következőképpen kaphatjuk:

$$n = A+B+O+AB$$

$$p' = 1-\sqrt{\frac{B+O}{n}}$$

$$q' = 1-\sqrt{\frac{A+O}{n}}$$

$$r' = \sqrt{\frac{O}{n}}$$

$$D = 1-(p'+q'+r')$$

$$p = p'\left(1+\frac{D}{2}\right)$$

$$q = q'\left(1+\frac{D}{2}\right)$$

$$r = 1-p-q.$$

(A szokásos $r=\left(r'+\frac{D}{2}\right)\left(1+\frac{D}{2}\right)$ képlettől való eltérést a későbbiekben ismertetett háromszögdiagram indokolja.)

Könnyen kiszámolható módszer még a [4], könyvben leírt *Dobson/Ikin-módszer*, melyet ott (indokolatlanul) *Fisher-módszer* néven említenek. A géngyakoriságok a *Dobson/Ikin-módszer* szerint a következők:

$$s = \sqrt{O}$$

$$t = \sqrt{A+O}$$

$$u = \sqrt{B+O}$$

$$v = t+u-s$$

$$p = \frac{t-s}{v}$$

$$q = \frac{u-s}{v}$$

$$r = 1-p-q$$

Észrevehetjük, hogy a *Dobson/Ikin-módszer* által szolgáltatott eredmények függetlenek az AB fenotípusok számától.

Fogalmazzuk meg a feladatot a következő módon. Keressük az ismert fenotípus-eloszláshoz valamilyen értelemben legközelebb eső *H—W-féle equilibriumot*. Ekkor a következő módon járhatunk el. Legyenek p, q és r az A, B és O gének,

AA, AO, BB, BO, OO, $\overline{AB}$ a megfelelő genotípusok gyakoriságai. A megfigyelt fenotípusok számát jelölje A, B, O és AB. Legyenek az $\mathbf{u}=(u_i)$ és $\mathbf{v}=(v_i)$ hatdimenziós vektorok.

$$\mathbf{u} = (AA, AO, BB, BO, OO, \overline{AB})$$

$$\mathbf{v} = (p^2, 2pr, q^2, 2qr, r^2, 2pq).$$

Legyen az $f=H(\mathbf{u}, \mathbf{v})$, ahol a H valamely rögzített, esetünkben a *Hellinger*-(arccos) *távolság:*

$$H(\mathbf{u}, \mathbf{v}) = \arccos \sum_{i=1}^{6} \sqrt{u_i v_i}.$$

Minthogy $AO=\frac{A}{n}-AA$, $BO=\frac{B}{n}-BB$, $r=1-p-q$, $(n=A+B+O+AB)$, az f függvény értékének kiszámításánál négy szabad paraméterünk adódik: AA, BB, p és q. Válasszuk az ABO géngyakoriságok becslésének az f négyváltozós függvény minimum helyének p_0 és q_0 koordinátáit, a genotípusok gyakoriságának becslésének pedig az AA_0, BB_0 koordinátákat, azaz

$$f(AA_0, BB_0, p_0, q_0) = \min \{f(AA, BB, p, q): AA, BB, p, q\}.$$

Ily módon nyerünk egy (p, q, r) paraméterű *H—W equilibriumot* és egy AA_0, $\frac{A}{n}-AA_0$, BB_0, $\frac{B}{n}-BB_0$, OO, $\overline{AB}$ genotípuseloszlást, melyek valamilyen távolság szerint a legközelebb fekszenek egymáshoz.

A *Hellinger-távolság* szerint kapott AA és BB genotípus-gyakoriságokra igaz a következő állítás.

Állítás: Tetszőleges rögzített p, q és r esetén a

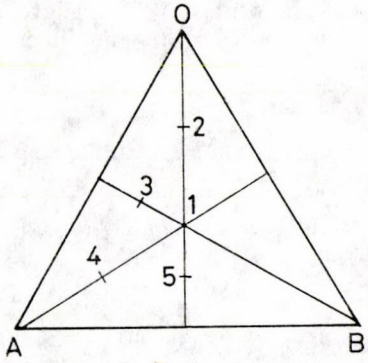
$$\mathbf{v} = (p^2, 2pr, q^2, 2qr, r^2, 2pq)$$

vektorhoz a *Hellinger-távolság* szerint legközelebb álló $\mathbf{u}=(AA, AO, BB, BO, OO, \overline{AB})$ vektorra $\frac{AA}{AO}=\frac{p^2}{2pr}$, $\left(\text{illetve } \frac{BB}{BO}=\frac{q^2}{2qr}\right)$.

Bizonyítás: A minimum helyre a $\frac{\partial f}{\partial AA}=0$ egyenlet, azaz

$$\frac{p}{2\sqrt{AA}} - \frac{\sqrt{2pr}}{2\sqrt{\frac{A}{n}-AA}} = 0$$

teljesül, ahonnan átrendezéssel a kívánt összefüggést kapjuk.



1. ábra

3.

Ebben a pontban módszerünket *Monte Carlo-vizsgálatnak* vetjük alá.

Annak a valószínűsége, hogy rögzített p, q és $r=1-p-q$ esetén egy populáció egy adott fenotípus-eloszlást mutat polinomiális elosztást követ [1].

Hogy tisztán lássunk, ábrázoljuk a (p, q, r) paraméterhármast az ABO szabályos háromszögben elhelyezkedő pontként. A szabályos háromszög magassága legyen egységnyi, a pontnak az A, B, O csúccsal szemközti oldaltól való távolsága legyen rendre p, q és r. Rögzítsük a paraméterhármast az ABO háromszög következő pontjaiban (1. táblázat, 1. ábra). Az így rögzített paraméterekkel $n=50$ létszámú populációkat generáltunk a megadott eloszlás szerint. A generált populációk száma $m=50$, 100, illetve 200 volt úgy, hogy a bővebb nem tartalmazta a szűkebbet. A 2—4. táblázatban a

1. TÁBLÁZAT

	p	q	r
1	1/3=0,333 333	1/3=0,333 333	1/3=0,333 333
2	1/6=0,166 666	1/6=0,166 666	2/3=0,666 666
3	5/12=0,416 666	1/6=0,166 666	5/12=0,416 666
4	2/3=0,666 666	1/6=0,166 666	1/6=0,166 666
5	5/12=0,416 666	5/12=0,416 666	1/6=0,166 666

$$V(p)+V(q)+V(r)=\frac{1}{m-1}\sum_{i=1}^{m}[(p_i-Ep)^2+(q_i-Eq)^2+(r_i-Er)^2]$$

(ahol E a várható érték) kifejezés értékét láthatjuk az 1, 2, 3, 4, 5 pontokban, a *Bernstein-, Dobson/Ikin-* és *Hellinger-módszerekkel*. A felhasznált függvényminimalizáló eljárás a *Powell-módszer* [5] volt.

A vizsgálat alapján úgy tűnik, hogy az általunk távolságmódszernek nevezett eljárás nem annyira efficiens, mint a *Bernstein-módszer*. A *Dobson/Ikin-módszer*, mint várható volt, jelentősen elmarad hatékonyságban az előbbi kettőtől.

2. TÁBLÁZAT

Bernstein-módszer

	50	100	200
1	0,011 398	0,010 014	0,010 990
2	0,007 592	0,005 079	0,005 587
3	0,009 596	0,008 417	0,008 597
4	0,008 404	0,011 450	0,010 163
5	0,011 598	0,009 929	0,010 026

3. TÁBLÁZAT

Dobson/Ikin-módszer

	50	100	200
1	0,018 404	0,016 000	0,015 734
2	0,010 724	0,006 633	0,006 649
3	0,012 532	0,011 309	0,012 089
4	0,017 707	0,019 177	0,017 826
5	0,016 941	0,016 097	0,017 580

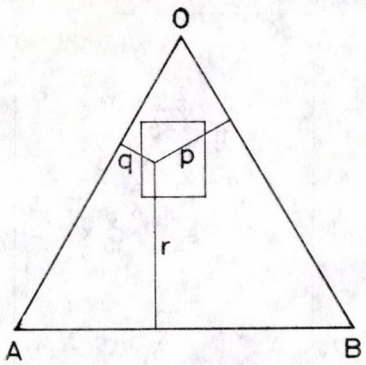
4. TÁBLÁZAT

Hellinger-módszer

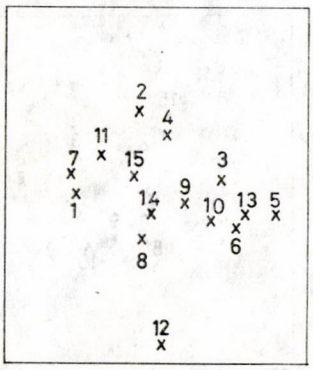
	50	100	200
1	0,011 770	0,010 363	0,011 273
2	0,007 574	0,005 131	0,005 846
3	0,009 888	0,008 522	0,008 738
4	0,008 718	0,012 008	0,010 590
5	0,011 917	0,010 250	0,010 412

4.

Végül módszerünket gyakorlati feladaton is kipróbálva LENGYEL IMRE [3] vizsgálataira fogunk hivatkozni. Az 5. táblázat adatai a *Kárpát-medencében* található különböző korú temetők adatainak összesítéséből származnak. (Az *A*, illetve *B* túlsúlyos csoport a kora magyar középkori minta bontása, ugyanúgy a *jugoszláv* neolitikus osztály is része az előtte álló neolitikus csoportnak). A géngyakoriságokat (6. táblázat) a fent leírt módon az *ABO* szabályos háromszögben ábrázolva nyerjük a 3—5. ábrát. (A 3—5. ábra a 2. ábrán látható téglalap nagyítása). A vizsgálat azt mutatja, hogy



2. ábra

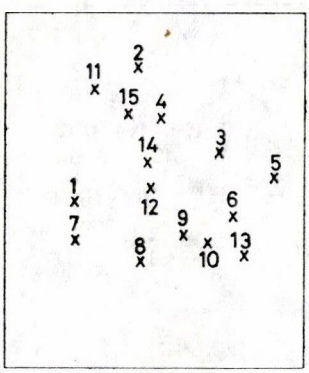


3. ábra
Bernstein-módszer

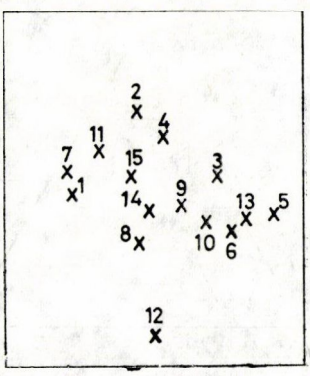
5. TÁBLÁZAT

		N	*A*	*B*	*O*	*AB*	*X*
1	RECENS BONCTERMI ANYAG	2042	852	343	624	159	64
2	ÚJKORI	915	253	141	330	81	110
3	KORA MAGYAR KÖZÉPKORI	1692	382	416	509	166	219
4	A TÚLSÚLYOS	816	219	157	265	66	109
5	B TÚLSÚLYOS	864	160	256	242	99	107
6	AVAR KORI	1651	403	477	450	180	141
7	„LANGOBARD" KORI	191	78	33	51	6	23
8	RÓMAI KORI	1082	356	235	257	92	142
9	KELTA KORI	90	28	24	25	7	6
10	VASKORI	59	16	16	15	5	7
11	SZKÍTA	48	15	6	16	5	6
12	BRONZKORI	870	226	155	213	193	83
13	RÉZKORI	398	102	128	103	30	35
14	NEOLITIKUS:	287	85	56	86	36	24
15	JUGOSZLÁV NEOLITIKUS	235	70	38	77	28	22
	ÖSSZESEN	9325	2796	2030	2679	960	860

a *Bernstein-* és a *távolság-módszer* által szolgáltatott eredmények majdhogynem egybeesnek, míg a *Dobson/Ikin-módszer* eredményei jelentősen különböznek az előbbiekétől.



4. ábra
Dobson/Ikin-módszer



5. ábra
Hellinger-módszer

6. TÁBLÁZAT

	Bernstein-módszer			*Dobson/Ikin-módszer*			*Hellinger-módszer*		
	p	*q*	*r*	*p*	*q*	*r*	*p*	*q*	*r*
1	0,3010	0,1363	0,5627	0,3018	0,1373	0,5609	0,3010	0,1363	0,5627
2	0,2322	0,1472	0,6206	0,2160	0,1278	0,6562	0,2313	0,1456	0,6232
3	0,2057	0,2203	0,5740	0,1933	0,2083	0,5984	0,2053	0,2199	0,5748
4	0,2260	0,1715	0,6024	0,2178	0,1623	0,6198	0,2258	0,1712	0,6030
5	0,1865	0,2678	0,5457	0,1676	0,2521	0,5803	0,1856	0,2673	0,5471
6	0,2153	0,2471	0,5376	0,2079	0,2402	0,5519	0,2152	0,2470	0,5379
7	0,2976	0,1257	0,5766	0,3151	0,1512	0,5337	0,2966	0,1222	0,5811
8	0,2777	0,1932	0,5291	0,2824	0,1990	0,5187	0,2776	0,1931	0,5292
9	0,2377	0,2070	0,5553	0,2457	0,2155	0,5388	0,2375	0,2067	0,5557
10	0,2287	0,2287	0,5426	0,2334	0,2334	0,5333	0,2287	0,2286	0,5427
11	0,2715	0,1385	0,5900	0,2505	0,1103	0,6392	0,2703	0,1356	0,5941
12	0,3020	0,2418	0,4562	0,2489	0,1797	0,5714	0,2999	0,2370	0,4631
13	0,2039	0,2506	0,5455	0,2153	0,2607	0,5240	0,2036	0,2503	0,5461
14	0,2611	0,1907	0,5482	0,2419	0,1681	0,5899	0,2604	0,1894	0,5503
15	0,2605	0,1662	0,5733	0,2380	0,1385	0,6235	0,2593	0,1639	0,5769

IRODALOM

[1] BERNSTEIN, F., „Fortgesetzte Untersuchungen aus der Theorie der Blutgruppen", *Zeitschrift für Induktionsabstammung- und Vereb-Lehre* **56** (1930) 233—273.

[2] KEMPHTHORNE, O., *An Introduction to Genetic Statistics* (John Wiley, New York, 1957).

[3] LENGYEL, I., *Palaeoserology* (Akadémiai Kiadó, Budapest, 1975).

[4] MOURANT, A. E., KOPEC, A. C., and DOMANIEWSKA—SOBCZAK, K., *The Distribution of the Human Blood Groups and Other Polymorphisms* (Oxford University Press, London, 1976).

[5] POWELL, M. J. O., "An efficient method for finding the minimum of a function of several variables without calculating derivatives", *The Computer Journal* 7 (1964) 155—162.

[6] RAO, C. R., "Cluster analysis applied to a study of race mixture in human populations", *Classification and Clustering ed. J. van Ryzin, Proceedings of an Advanced Seminar Conducted by the Mathematics Research Center the University of Wisconsin at Madison* (Academic Press, New York, 1977).

(Beérkezett: 1980. január 23.)

(Újra beérkezett: 1980 október 27.)

JUHÁSZ FERENC
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST, ÚRI U. 49.

ON THE HELLINGER DISTANCE

F. JUHÁSZ

The *Hellinger* (arccos) *distance* is the subject of the article with the purpose of the demonstration in what an extent does it represent the differences among populations. Employing the *Hellinger distance* to the estimation of gene frequencies, a result very similar to that of the *maximum likelihood (Bernstein) method* can be gained. Besides, I should like to call the attention to the fact that in case of palaeoseroloical examinations the *Dobson/Ikin method* mentioned by MOURANT et al. as *Fisher method,* cannot be justified.

Alkalmazott Matematikai Lapok 6 (1980) 237—286

TAPASZTALATI FÜGGVÉNYEK SIMÍTÁSA

NYÍRI ANDRÁS

Budapest

Egyenközű pontokban adott közelítő függvényértékek simítására szolgáló eljárást mutat be a a dolgozat. A simított értékrendszer előállítása úgy történik, hogy annak a simítandótól való eltérése és az r-edik differenciája valamennyi pontot tekintve egyidejűleg legyen minimális.

1. Bevezetés

Megfigyelésekből, mérésekből kapott adatok feldolgozása során nehézségekbe ütközünk, mivel az így adott számok eredendően hibásak. Különösen nagy problémát okoz, ha az ilyen, pontokban ismert értékek alapján a pontbeli differenciálhányados meghatározására is szükség van.

Diszkrét pontokban megadott, pl. empirikus úton nyert, közelítő függvényértékeket szokásos a legkisebb négyzetek elve alapján a „legjobban közelítő" polinomokkal helyettesíteni. A polinomok fokszámának megválasztása igen kényes probléma. Ha ugyanis azt túl kicsinek választjuk, akkor az éppen ezért nem közelít megfelelően, ha pedig túl nagynak vesszük, akkor az irregularitásokat nem szűrjük ki.

Abban az esetben természetesen, ha az adatokról tudjuk, hogy azok elméletileg milyen függvény közelítő értékei, akkor magát az elméleti függvényt választjuk alapul és annak paramétereit határozzuk meg valamely, a hibát minimalizáló numerikus eljárással. Ez utóbbi esettől eltekintve mindenképpen önkényesen járunk el, ha a pontrendszeren át polinomot fektetünk — függetlenül attól, hogy milyen szerencsével választjuk meg annak fokszámát. Igen nagy hátránya még a polinomokkal való simításnak az is, hogy az adatrendszert a szélein nem lehet ily módon simítani.

A dolgozatban a polinomokat elkerülő eljárást mutatunk be, amely az említett önkényességtől — éppen ezért — mentes. A módszert WHITTAKER [8] ajánlotta több mint 50 évvel ezelőtt. WHITTAKER módszerének rövid ismertetése után általánosítjuk azt. Az általánosított simítási feladat megfogalmazása után, annak olyan megoldási módszerét dolgozzuk ki, amely a megoldást lényegesen egyszerűbben szolgáltatja, s amelyhez nem lesz szükség előre kidolgozott táblázatok használatára. A simítás mértékének megválasztása tetszőleges lehet. A simítási eljárás nem igényel számítógépet, hanem egyszerű számológéppel is elvégezhető.

2. A Whittaker-féle simítás

Legyen az $f(x)$ függvény $x_k (k=1, 2, \ldots, n)$ egyenközű helyeken felvett értékeinek közelítő értékei a g_k számok. Az eltérésekről feltesszük, hogy véletlenszerűek, amelyek pl. mérési pontatlanságok következményei. Az (x_k, g_k) pontokra illeszkedő görbe nem megfelelően sima, de van alapja annak a feltevésnek, hogy pontosabb mérésekkel olyan adatrendszert kapnánk, melyekre a rárajzolható görbe sima lenne.

Kézenfekvő az a kérdés, hogy milyen módon nyerhetünk a g_k közelítő adatok alapján sima, a helyes $f(x_k)$ értékeket jobban megközelítő értékrendszert.

A simaság bizonytalan jelentését WHITTAKER [8] a következőképpen tette megfoghatóvá. Értelmezzük az x_k $(k=1, 2, \ldots, n)$ pontokban adott f_k számsorozat simaságát a belőlük alkotható — mondjuk — harmadik differenciák négyzeteinek összegével:

$$\sum_{k=1}^{n-3} (\Delta^3 f_k)^2. \tag{2.1}$$

A mérési hibákról tegyük fel, hogy normális valószínűségeloszlási függvényt követnek. WHITTAKER valószínűségszámítási meggondolások alapján arra az eredményre jutott, hogy a mérés eredményeként nyert g_k adatrendszer alapján a helyes értékeket legnagyobb valószínűséggel megközelítő értékrendszert az

$$S = \varepsilon \sum_{k=1}^{n} (g_k - f_k)^2 + \sum_{k=1}^{n-3} (\Delta^3 f_k)^2 \tag{2.2}$$

összeget minimalizáló f_k-k megkeresésével kapjuk meg. Az $\varepsilon > 0$ simítási paraméter értékét a simítatlan értékrendszer előzetes vizsgálata alapján kell megválasztani.

Az S összeg minimumát az

$$\varepsilon f_k - \Delta^6 f_{k-3} = \varepsilon g_k \tag{2.3}$$

differenciaegyenlet megoldásából nyert f_k értékrendszeren veszi fel. A differenciaegyenlet megoldása:

$$f_k = -\varepsilon \left[\frac{E^3}{(E-1)^6 - \varepsilon E^3} \right] g_k, \tag{2.4}$$

ahol E az eltolás operátora (a Δ differenciaoperátor és az eltolás operátora között a kapcsolat: $E = 1 + \Delta$). A megoldás a következő simító formulára vezet:

$$f_k = k_0 g_k + k_1 (g_{k+1} + g_{k-1}) + k_2 (g_{k+2} + g_{k-2}) + \ldots, \tag{2.5}$$

ahol a $k_0, k_1, k_2, \ldots$ együtthatók kiszámítására szolgáló összefüggések az idézett műben találhatók.

Ezen együtthatók értékeit ε néhány konkrét értékére kiszámítva táblázatba foglaltan ugyanott találhatjuk 4 tizedes pontosságig.

A (2.5) simító formula alkalmazásához szükség van a g_k értékrendszer kiterjesztésére az x_1 előtti és az x_n utáni $n-1$ helyen. A kiterjesztést azon feltétellel nyerjük, hogy az eredeti határokon túli g_k értékeket tartalmazó harmadik differenciák eltűnését írjuk elő. Kikötjük továbbá azt, hogy ha $k < 1$ és ha $k > n$, akkor $f_k = g_k$. A kiterjesztett értékrendszer kiszámítására a

$$g_k = j_1 g_{k-1} + j_2 g_{k-2} + \ldots \tag{2.6}$$

formula szolgál. A j együtthatókat a (2.4) egyenlethez hasonló egyenlet megoldásából nyerjük. A j együtthatókat ε néhány értékére szintén az említett műben találjuk.

A simítás mértékére a következő meggondolással jutunk. Tegyük fel, hogy a g-k valószínű hibája p, akkor a harmadik differenciáké $p\sqrt{1^2+3^2+3^2+1^2}=p\sqrt{20}$. Továbbá mivel

$$f_0 = \sum k_m g_m \quad \text{és} \quad \Delta^3 f_0 = \sum (\Delta^3 k_m) g_m,$$

azért $\Delta^3 f_0$ valószínű hibája $p\sqrt{\sum (\Delta^3 k_m)^2}$. WHITTAKER végül is a $\sqrt{\sum (\Delta^3 k_m)^2}/\sqrt{20}$ hányadost nevezi simítási együtthatónak, amely az ε simítási paramétertől függ.

WHITTAKER vázolt eljárása alkalmazásakor kötve vagyunk a simítási paraméter konkrét értékeihez, illetőleg minden egyes ε-ra táblázatot kell kidolgozni. A következőkben olyan simító eljárást mutatunk be, amelyet a simítás általánosított differenciaegyenlete más úton való megoldása révén nyerünk. A simítás végrehajtásához nem lesz szükség előkészített táblázatokra, sem az intervallum háromszorosra való kiterjesztésére.

3. A simítás differenciaegyenlete

Legyen adva az x_k $(k=1, 2, \ldots, n)$ egyenközű helyeken a g_k simítandó értékrendszer, amelynek a helyes értéktől való eltérése normális eloszlású. Keressük az f_k $(k=1, 2, \ldots, n)$ értékrendszert úgy, hogy az a legnagyobb valószínűséggel közelítse a helyes értékrendszert.

A (2.2) összegben szereplő harmadik differenciák helyett vehetünk r-edik differenciákat, mivel a (2.2) feltétel levezetésénél a differencia rendje nem volt kihasználva. Feladatunk tehát az

$$S_r = \varepsilon \sum_{k=1}^{n} (g_k - f_k)^2 + \sum_{k=1}^{n-r} (\Delta^r f_k)^2 \tag{3.1}$$

összeget minimalizáló f_k $(k=1, 2, \ldots, n)$ számsorozat előállítása $\varepsilon>0$ esetén.

Az S_r összeg minimumának szükséges feltétele az, hogy a változói szerinti első differenciálhányadosok eltűnjenek:

$$\frac{\partial S_r}{\partial f_k} = -2\varepsilon(g_k - f_k) + 2(-1)^r \Delta^{2r} f_{k-r} = 0.$$

Ebből az f_k értékeire a következő differenciaegyenletet nyerjük:

$$\varepsilon f_k + (-1)^r \Delta^{2r} f_{k-r} = \varepsilon g_k \quad (k = r+1, r+2, \ldots, n-r). \tag{3.2}$$

A (3.2) egyenletrendszer alapján még nem határozhatók meg az $f_1, f_2, \ldots, f_r$ és az $f_{n-r+1}, f_{n-r+2}, \ldots, f_n$ értékek. A hiányzó $2r$ számú érték meghatározása végett terjesszük ki a g_k adatrendszert úgy, hogy

$$\Delta^r g_k = 0 \quad \text{legyen és} \quad f_k = g_k \quad \text{teljesüljön,} \tag{3.3}$$

ha $k=0, -1, -2, \ldots, -r+1$ és $k=n+1, n+2, \ldots, n+r$. Ez a kiterjesztés azt jelenti, hogy az eredeti intervallumon *kívül* az adatrendszerünket egy $r-1$-edfokú parabolával folytatjuk.

A bővítéssel elértük, hogy valamennyi, az eredeti intervallumhoz tartozó k-ra azonos alakú egyenletet kaptunk:

$$\varepsilon f_k + (-1)^r \Delta^{2r} f_{k-r} = \varepsilon g_k \quad (k = 1, 2, \ldots, n). \tag{3.4}$$

A (3.4) differenciaegyenlet (3.3) peremfeltétel melletti megoldása szolgáltatja a g_k közelítő értékek alapján meghatározható azon f_k számokat, amelyek az $f(x)$ függvénynek az x_k helyeken felvett értékeit a legnagyobb valószínűséggel közelítik meg.

A (3.4) differenciaegyenlet adott peremfeltételek melletti megoldását a homogén egyenlet ($g_k=0$) általános és az inhomogén egyenlet egy partikuláris megoldása összegeként állítjuk elő.

4. A homogén differenciaegyenlet megoldása

Tekintsük a következő homogén, lineáris, állandó együtthatós differenciaegyenlet-rendszert:

$$\varepsilon f_k + (-1)^r \Delta^{2r} f_{k-r} = 0 \quad (k = 1, 2, \ldots, n). \tag{4.1}$$

Ismeretes, hogy ennek megoldása λ^k alakú kifejezések lineáris kombinációjaként állítható elő [2]. Az $f_k=\lambda^k$ helyettesítéssel megkapjuk a differenciaegyenlet karakterisztikus egyenletét:

$$(\lambda-1)^{2r} + (-1)^r \varepsilon \lambda^r = 0. \tag{4.2}$$

Gyökvonással a karakterisztikus egyenlet a következő másodfokú egyenletekre redukálható:

$$\lambda^2 - (2+\varepsilon_j)\lambda + 1 = 0,$$

ahol

$$\varepsilon_j = \begin{cases} \sqrt[r]{\varepsilon}\, e^{i\frac{2\pi}{r}j}, & \text{ha } r \text{ páratlan} \\ \sqrt[r]{\varepsilon}\, e^{i\frac{\pi+2\pi j}{r}}, & \text{ha } r \text{ páros} \end{cases} \qquad (j = 0, 1, \ldots, r-1). \tag{4.3}$$

A (4.2) egyenletnek $2r$ számú különböző gyöke van. Ha r páratlan akkor a $j=0$-hoz tartozó két gyök valós. Könnyen belátható, hogy az ε_j-hez tartozó gyökök az $\bar{\varepsilon}_j=\varepsilon_{r-j}$-hez tartozóknak komplex konjugáltjai, ha r páratlan, feltéve, hogy $j\neq 0$. Legyenek az ε_j-hez tartozó gyökök λ_1 és λ_2. Akkor $\lambda_1+\lambda_2=2+\varepsilon_j$ és $\lambda_1\lambda_2=1$. De ez esetben $\overline{\lambda_1+\lambda_2}=2+\bar{\varepsilon}_j$ és $\overline{\lambda_1\lambda_2}=1$, ezért $\lambda_1'=\bar{\lambda}_1$ és $\lambda_2'=\bar{\lambda}_2$ az $\bar{\varepsilon}_j$-hoz tartozó két gyök.

Ha r páros, akkor minden gyök komplex és páronként egymás konjugáltjai.

Mivel a karakterisztikus egyenlet gyökei mind különbözőek, azért a homogén differenciaegyenlet általános megoldását azok lineáris kombinációja előállítja [2]. Az általános megoldás a következő:

$$\sum_{l=1}^{2r} c_l' \lambda_l^k. \tag{4.4}$$

A simításhoz valós megoldást keresünk. Kihasználjuk azt a körülményt, hogy a karakterisztikus egyenlet komplex gyökei konjugált párjukkal együtt fordulnak

elő. Legyenek λ_p és λ_q egymás konjugáltjai, azaz

$$\lambda_p = \varrho(\cos\alpha + i\sin\alpha), \quad \lambda_q = \varrho(\cos\alpha - i\sin\alpha).$$

Ekkor a $c'_p\lambda_p^k + c'_q\lambda_q^k$ összeget $c_p\varrho^k\cos k\alpha + c_q\varrho^k\sin k\alpha$ alakra hozhatjuk, ahol c_p és c_q már valósak.

Ezekkel a (4.1) egyenletrendszer általános valós megoldása:

$$f_k^* = \sum_{l=1}^{2r} c_l f_k^{l*}, \tag{4.5}$$

ahol $f_k^{l*} = \begin{cases} \lambda_l^k, & \text{ha } \lambda_l \text{ valós} \\ \varrho_l^k \begin{matrix}\sin \\ \cos\end{matrix} k\alpha_l, & \text{ha } \lambda_l \text{ komplex.} \end{cases}$

5. Az inhomogén differenciaegyenlet megoldása

A (3.4) inhomogén differenciaegyenlet egy partikuláris megoldása előállításához egy valamivel általánosabb feladat megoldását fogjuk alkalmazni.

Tekintsük a következő n ismeretlenes, lineáris algebrai egyenletrendszert, amelynek együtthatómátrixa $2r+1$ szélességű *Jacobi-féle szalagmátrix:*

$$\sum_{q=-r}^{r} a_k^{(q)} u_{k+q} = b_k \quad (k = 1, 2, \ldots, n) \quad 1 \leqq r \leqq n-1. \tag{5.1}$$

Az (5.1) egyenletrendszer megoldását a következő alakban állítjuk elő, amely tulajdonképpen egy rekurziós formula:

$$u_k = \beta_k + \sum_{p=1}^{r} \alpha_k^{(p)} u_{k+p}. \tag{5.2}$$

Az $\alpha_k^{(p)}$ és β_k együtthatókra a következő eljárással szintén rekurziós formulákat nyerünk. Írjuk fel az (5.2) kifejezést rögzített k esetén u_{k-1}, u_{k-2}, ... u_{k-r}-re. Helyettesítsük u_{k-1} ezen alakját u_{k-2}-be. Az u_{k-1} és u_{k-2} előbb nyert alakját helyettesítsük u_{k-3}-ba. Folytassuk az eljárást u_{k-r}-ig bezárólag. Ily módon $u_{k-1}, u_{k-2}, \ldots, u_{k-r}$ kifejezései $u_k, u_{k+1}, \ldots, u_{k+r}$-et, továbbá az $\alpha_l^{(p)}$ és a $\beta_l (l<k)$ együtthatókat tartalmazzák. Helyettesítsük $u_{k-1}, u_{k-2}, \ldots, u_{k-r}$-et az (5.1) egyenletrendszer rögzített k-adik sorába, majd ennek (5.2)-vel való összehasonlításából megkapjuk az $\alpha_k^{(p)}$ $(p=1, 2, \ldots, r)$ és a β_k kiszámítására szolgáló rekurzív formulákat, amelyek az ismert $a_k^{(q)}$ és b_k együtthatókon kívül csak k-nál kisebb indexű α-kat és β-kat tartalmaznak.

Az eredmény a következőkben foglalható össze. Az $\alpha_k^{(p)}$ és β_k kiszámolására szolgáló rekurzív formulákat a következő $(2r+2)(r+1)$-es mátrixból képzett bizonyos determinánsok segítségével fogjuk megadni:

$$\mathbf{M}_k = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & r & r+1 & r+2 & 2r+1 & 2r+2 \end{matrix} \\ \left[\begin{array}{ccc:ccc:cc} a_k^{(-r)} & a_k^{(-r+1)} & a_k^{(-r+2)} & a_k^{(-1)} & a_k^{(0)} & a_k^{(1)} & a_k^{(r)} & -b_k \\ -1 & \alpha_{k-r}^{(1)} & \alpha_{k-r}^{(2)} & \alpha_{k-r}^{(r-1)} & \alpha_{k-r}^{(r)} & 0 & 0 & \beta_{k-r} \\ 0 & -1 & \alpha_{k-r+1}^{(1)} & \alpha_{k-r+1}^{(r-2)} & \alpha_{k-r+1}^{(r-1)} & \alpha_{k-r+1}^{(r)} & 0 & \beta_{k-r+1} \\ 0 & 0 & -1 & \alpha_{k-r+2}^{(r-3)} & \alpha_{k-r+2}^{(r-2)} & \alpha_{k-r+2}^{(r-1)} & 0 & \beta_{k-r+2} \\ \hdashline 0 & 0 & 0 & -1 & \alpha_{k-1}^{(1)} & \alpha_{k-1}^{(2)} & 0 & \beta_{k-1} \end{array}\right] \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ \\ r+1 \end{matrix} \end{array} \tag{5.3}$$

ahol $\alpha_j^{(p)}=0$ és $\beta_j=0$, ha $j<1$ bármely p-re, továbbá $\alpha_j^{(p)}=0$, ha $j+p>n$ és $\beta_j=0$, ha $j>n$.

Tegyük fel, hogy $u_j=\beta_j+\sum_{p=1}^{r}\alpha_j^{(p)}u_{j+p}$ igaz minden $j\leqq k-1$ esetén. Jelölje $\mathbf{u}_k$ a következő $2r+2$ elemű oszlopvektort:

$$\mathbf{u}_k=\begin{bmatrix} u_{k-r}\\ \vdots\\ u_{k-1}\\ u_k\\ u_{k+1}\\ \vdots\\ u_{k+r}\\ 1\end{bmatrix}. \tag{5.4}$$

Az $\mathbf{u}_k$ vektor felső $r+1$ eleme alkossa az $\mathbf{u}_L=\mathbf{u}_{Lk}$ $r+1$ elemű oszlopvektort és a maradék $r+1$ elem az $\mathbf{u}_R=\mathbf{u}_{Rk}$ oszlopvektort. Ekkor az előbbi feltevésünk és az (5.1) egyenletrendszer alapján nyilvánvaló, hogy

$$\mathbf{M}_k\mathbf{u}_k=\mathbf{0} \tag{5.5}$$

Jelöljük az $\mathbf{M}_k$ mátrix bal oldali első $r+1$ oszlopából képzett mátrixot $\mathbf{L}$-lel, a maradék $r+1$ oszlop alkotta márixot $\mathbf{R}$-rel. A következő azonosság azonnal felismerhető:

$$\mathbf{M}_k\mathbf{u}_k\equiv\mathbf{L}\mathbf{u}_L+\mathbf{R}\mathbf{u}_R=\mathbf{0}. \tag{5.6}$$

Az $\mathbf{L}\mathbf{u}_L=-\mathbf{R}\mathbf{u}_R$ egyenletrendszernek $r+1$ elemű megoldásvektorából r számút ismerünk. Az utolsó, az u_k elem kifejezhető az u_{k+p}, $p=1, 2, \ldots, r$ és 1 lineáris kombinációjaként:

$$u_k=\sum_{p=1}^{r}-\frac{D_k^{(p)}}{D_k^{(0)}}u_{k+p}-\frac{D_k^{(b)}}{D_k^{(0)}}, \tag{5.7}$$

ahol a $D_k^{(p)}$ $(p=1, 2, \ldots, r)$ determináns elemeit az $\mathbf{L}$ mátrix első r oszlopvektorának és az $\mathbf{L}$ utolsó oszlopa helyébe beírt, az $\mathbf{R}$ mátrix p-edik oszlopvektorának elemei alkotják (az $\mathbf{R}$ nulladik oszlopa definíció szerint legyen az $\mathbf{L}$ utolsó oszlopa). A $D_k^{(b)}$ determináns elemeit úgy kapjuk, hogy $\mathbf{L}$ utolsó oszlopa helyébe $\mathbf{R}$ utolsó oszlopát helyezzük. Ezek után

$$\alpha_k^{(p)}=-\frac{D_k^{(p)}}{D_k^{(0)}}\quad\text{és}\quad\beta_k=-\frac{D_k^{(b)}}{D_k^{(0)}} \tag{5.8}$$

választásával u_k-t szintén (5.2) alakban állítottuk elő. Az (5.2) alak helyességét $k=1$-re azonnal beláthatjuk, ugyanis

$$\alpha_1^{(p)}=-\frac{a_1^{(p)}}{a_1^{(0)}},\quad\beta_1=\frac{b_1}{a_1^{(0)}}. \tag{5.9}$$

A most kapott eredményt alkalmazhatjuk a (3.4) differenciaegyenlet partikuláris megoldása előállításához. Ehhez az (5.1) egyenletrendszert a következő együtthatók-

kal kell megoldani:

$$a_k^{(q)} = \begin{cases} (-1)^r(-1)^{r-q}\binom{2r}{r+q}, & \text{ha} \quad q \neq 0 \text{ és } 1 \leqq q+k \leqq n \\ \binom{2r}{r}+\varepsilon, & \text{ha} \quad q = 0 \\ 0, & \text{ha} \quad q+k \leqq 0 \text{ és, ha } q+k \geqq n+1 \end{cases} \tag{5.10}$$

és $b_k = \varepsilon g_k$ $(k = 1, 2, \ldots, n)$

6. A differenciaegyenlet megoldása

A (3.4) differenciaegyenlet megoldása az inhomogén egyenlet u_k $(k = 1, 2, \ldots, n)$ partikuláris megoldása és a (4.1) homogén egyenlet általános megoldásának összege:

$$f_k = u_k + \sum_{l=1}^{2r} c_l f_k^{l*} \quad (k = 1, 2, \ldots, n). \tag{6.1}$$

A c_l együtthatókat a (3.3) peremfeltétel figyelembevételével határozzuk meg:

$$\sum_{l=1}^{2r} c_l f_k^{l*} = f_k (= g_k), \tag{6.2}$$

$$\text{ha} \quad k = 0, -1, \ldots, -r+1; \quad n+1, \ldots, n+r.$$

A c_l-re nézve $2r$ ismeretlenes (6.2) egyenletrendszer megoldásával a (6.1) kifejezés feladatunk megoldását szolgáltatja. (A (6.2) alatti egyenletrendszer megoldhatósági kérdésével nem foglalkozunk.)

7. A simítás mértéke

A WHITTAKER által javasolt simítási együtthatót egyszerű meggondolásokkal az ε simítási paraméter függvényeként állítjuk elő.

Tekintsük a g_k-kat valószínűségi változóknak, melyek szórását jelöljük σ_g-vel, azaz $D^2(g_k) = \sigma_g^2$. Hasonlóan legyen $D^2(f_k) = \sigma_f^2$. A simítás mértéke legyen a simított és a simítatlan értékekből képezhető r-edik differenciák szórásainak hányadosa:

$$s = \sqrt{\frac{D^2(\Delta^r f_k)}{D^2(\Delta^r g_k)}}. \tag{7.1}$$

Ismeretes, hogy $\Delta^r g_k = \sum_{j=0}^{r} (-1)^{r-j}\binom{r}{j} g_{k+j}$, aminek felhasználásával:

$$D^2(\Delta^r g_k) = \sigma_g^2 \sum_{j=0}^{r}\binom{r}{j}^2 = \sigma_g^2\binom{2r}{r}.$$

Hasonlóképpen kapjuk, hogy $D^2(\Delta^r f_k)=\sigma_f^2\binom{2r}{r}$. A kapcsolatot g_k és f_k között a (3.4) differenciaegyenlet adja, amelynek alapján

$$\varepsilon^2\sigma_g^2 = D^2(\varepsilon g_k) = D^2[\varepsilon f_k+(-1)^r\Delta^{2r}f_{k-r}] = \tag{7.2}$$

$$= \sigma_f^2\left\{\sum_{\substack{j=0\\ j\neq r}}^{2r}\binom{2r}{j}^2+\left[\binom{2r}{r}+\varepsilon\right]^2\right\}.$$

A (7.2) összefüggés felhasználásával megkapjuk a simítási együtthatót mint a simítási paraméter függvényét:

$$s = \frac{\varepsilon}{\sqrt{\binom{4r}{2r}-\binom{2r}{r}^2+\left[\binom{2r}{r}+\varepsilon\right]^2}}. \tag{7.3}$$

A simítás mértéke a simítási paraméter megválasztásától függ. Abban az esetben, ha a kiinduló adatrendszer szórása kicsi, vagyis viszonylag pontos, akkor s-et és így ε-t sem szükséges kicsinek választani.

8. Összehasonlítás más simítási eljárással

A bemutatott simítási eljárást érdekes megvilágításba helyezi pl. az ún. ötpontos simító formulákkal való összehasonlítás. Az ötpontos simító formula alakja szimmetriaokokból a következő:

$$f_k = g_k+\frac{1}{C}\Delta^4 g_{k-2}. \tag{8.1}$$

Az ötpontos formulával elérhető simítás mértéke a C állandó megválasztásától függ. Az s simítási együttható a korábbiak felhasználásával így írható:

$$s_5 = \frac{\sigma_f}{\sigma_g} = \sqrt{\frac{34+(6+C)^2}{C^2}}. \tag{8.2}$$

Az s_5-nek minimuma van $C=-\frac{70}{6}$-nál és itt $s_5=s_c=0{,}696\,932$. A legjobban simító ötpontos formula tehát:

$$f_k = g_k-\frac{6}{70}\Delta^4 g_{k-2}. \tag{8.3}$$

A (8.3) formula alkalmazásához k előtti és utáni két-két érték ismerete szükséges. A szélekre is előállíthatunk ötpontos formulákat ortogonális polinomok szerinti sorbafejtés segítségével. Öt szomszédos ponton átmenő harmadfokú parabola esetén

a simító formulák:

$$
\begin{aligned}
f_{k-2} &= g_{k-2} - \frac{1}{70}\Delta^4 g_{k-2} \\
f_{k-1} &= g_{k-1} + \frac{4}{70}\Delta^4 g_{k-2} \\
f_k &= g_k - \frac{6}{70}\Delta^4 g_{k-2} \\
f_{k+1} &= g_{k+1} + \frac{4}{70}\Delta^4 g_{k-2} \\
f_{k+2} &= g_{k+2} - \frac{1}{70}\Delta^4 g_{k-2}.
\end{aligned} \tag{8.4}
$$

A (8.4) ötpontos simító formulákban a még nem vizsgált esetekben $C=-70$ és $\frac{70}{4}$. Az első esetben a C konstanshoz tartozó $s_5<1$, de a másodikban már $s_5>1$. Ebből pedig az következik, hogy az $s_5>1$ esetén a simító formula alkalmazásával nem simított értéket kapunk, hanem a kiindulónál még inkább szórtat. Következésképpen az ötpontos formulának a széleken való alkalmazásakor nem érhetjük el a kívánt célt, ismételt alkalmazásával még kevésbé.

A simításhoz figyelembe vett pontok számának növelésével sem tudjuk csökkenteni a simítási együttható értékét. A 7 pontos formulákat az $r=3$-hoz tartozó differenciaegyenlet simítási együtthatójával lehet összevetni:

$$s_7 = \sqrt{\frac{524+(20+C)^2}{C^2}}. \tag{8.5}$$

Ennek a függvénynek $C=-\frac{924}{20}$-nál van a minimuma, értéke pedig (0,753060) nagyobb, mint s_c.

9. Simítás $r=2$ esetében

A második differenciák szórásának csökkentésével hatékony simítást érhetünk el. A megoldandó (3.4) differenciaegyenlet és a (3.3) peremfeltétel $r=2$ esetében a következő:

$$
\begin{aligned}
&\varepsilon f_k + \Delta^4 f_{k-2} = \varepsilon g_k \quad (k = 1, 2, \ldots, n) \\
&\Delta^2 g_k = 0,\ f_k = g_k \quad (k = -1, 0, n+1, n+2).
\end{aligned} \tag{9.1}
$$

A homogén differenciaegyenlet megoldása:

$$f_k^* = \sum_{l=1}^{4} c_l f_k^{l*} = c_1 \varrho^k \cos k\alpha + c_2 \varrho^k \sin k\alpha + c_3 \varrho^{-k} \cos k\alpha + c_4 \varrho^{-k} \sin k\alpha, \tag{9.2}$$

ahol

$$\varrho = \left\{\left[\frac{\sqrt{\varepsilon}}{2}+\sqrt{\frac{\sqrt{\varepsilon}}{4}\sqrt{\varepsilon+16}}\sin\left(\frac{\alpha'}{2}+45°\right)\right]^2+\right.$$
$$\left.+\left[1+\sqrt{\frac{\sqrt{\varepsilon}}{4}\sqrt{\varepsilon+16}}\cos\left(\frac{\alpha'}{2}+45°\right)\right]^2\right\}^{1/2},$$

$$\tan\alpha = \frac{\frac{\sqrt{\varepsilon}}{2}+\sqrt{\frac{\sqrt{\varepsilon}}{4}\sqrt{\varepsilon+16}}\sin\left(\frac{\alpha'}{2}+45°\right)}{1+\sqrt{\frac{\sqrt{\varepsilon}}{4}\sqrt{\varepsilon+16}}\cos\left(\frac{\alpha'}{2}+45°\right)}, \quad \tan\alpha' = \frac{\sqrt{\varepsilon}}{4}.$$

Az inhomogén egyenletrendszer most a következő:

$$\sum_{q=-2}^{2} a_k^{(q)} u_{k+q} = b_k, \tag{9.3}$$

ahol

$$a_k = \begin{cases} (-1)^{2-q}\binom{4}{2+q}, & \text{ha} \quad q = -2, -1, 1, 2 \\ \binom{4}{2}+\varepsilon, & \text{ha} \quad q = 0 \end{cases}$$

$$b_k = \varepsilon g_k.$$

Az $\mathbf{M}_k$ mátrix $r=2$ esetén:

$$\mathbf{M}_k = \begin{bmatrix} 1 & -4 & 6+\varepsilon & -4 & 1 & -\varepsilon g_k \\ -1 & \alpha_{k-2}^{(1)} & \alpha_{k-2}^{(2)} & 0 & 0 & \beta_{k-2} \\ 0 & -1 & \alpha_{k-1}^{(1)} & \alpha_{k-1}^{(2)} & 0 & \beta_{k-1} \end{bmatrix} \tag{9.4}$$

Az $\mathbf{M}_k$-ból képzett determinánsok, amelyekkel a rekurziós formulák előállíthatók, a következők:

$$\begin{aligned} D_k^{(0)} &= \alpha_{k-2}^{(1)}\alpha_{k-1}^{(1)}+\alpha_{k-2}^{(2)}-4\alpha_{k-1}^{(1)}+6+\varepsilon \\ D_k^{(1)} &= \alpha_{k-2}^{(1)}\alpha_{k-1}^{(2)}-4\alpha_{k-1}^{(2)}-4 \\ D_k^{(2)} &= 1 \\ D_k^{(b)} &= \alpha_{k-2}^{(1)}\beta_{k-1}+\beta_{k-2}-4\beta_{k-1}-\varepsilon g_k. \end{aligned} \tag{9.5}$$

Maguk a rekurziós formulák pedig a következők:

$$\begin{aligned} \alpha_k^{(1)} &= \frac{4-(\alpha_{k-2}^{(1)}-4)\alpha_{k-1}^{(2)}}{6+\varepsilon+(\alpha_{k-2}^{(1)}-4)\alpha_{k-1}^{(1)}+\alpha_{k-2}^{(2)}}, \quad \alpha_1^{(1)} = \frac{4}{6+\varepsilon} \\ \alpha_k^{(2)} &= \frac{-1}{6+\varepsilon+(\alpha_{k-2}^{(1)}-4)\alpha_{k-1}^{(1)}+\alpha_{k-2}^{(2)}}, \quad \alpha_1^{(2)} = \frac{-1}{6+\varepsilon} \\ \beta_k &= \frac{\varepsilon g_k-(\alpha_{k-2}^{(1)}-4)\beta_{k-1}-\beta_{k-2}}{6+\varepsilon+(\alpha_{k-2}^{(1)}-4)\alpha_{k-1}^{(1)}+\alpha_{k-2}^{(2)}}, \quad \beta_1 = \frac{\varepsilon g_1}{6+\varepsilon}. \end{aligned} \tag{9.6}$$

Ezek felhasználásával az inhomogén egyenlet megoldása:

$$u_k = \beta_k + \alpha_k^{(1)} u_{k+1} + \alpha_k^{(2)} u_{k+2}. \tag{9.7}$$

A simítási együttható esetünkben:

$$s = \frac{\varepsilon}{\sqrt{34+(6+\varepsilon)^2}} \tag{9.8}$$

10. Példák

Az eljárás végrehajtását példákon mutatjuk be. A simítandó értékrendszert úgy választjuk meg, hogy egy pontosan kiszámolható függvénynek pontatlan értékeit vesszük, így az eljárás hatékonysága is megállapítható lesz. A példához az $r=2$ értéket választottuk.

Tekintsük az e^x függvényt a [0, 1] intervallumban. Vegyük a lépésközt 0,1-nek. Itt a függvény értékei: $e^{\frac{k-1}{10}}$, $(k=1, \ldots, 11)$. A simítandó g_k értékek legyenek $e^{\frac{k-1}{10}}$ két tizedesjegyre kerekített értékei (1. táblázat). A $k=0, -1$ és 12, 13 helyekre g_k értékeit a (3.3) feltétel alapján határozzuk meg: $\Delta^2 g_0 = 0$-ból $g_0 = 2g_1 - g_2$, stb. Az ε simítási paraméter értékét 3-nak választjuk. $\alpha_k^{(1)}$, $\alpha_k^{(2)}$ és β_k értékeit a (9.6), majd u_k-t a (9.7) képlet alapján számoljuk ki. A homogén egyenlet általános megoldásához ϱ és α értékeit a (9.2) képletekből kiszámítjuk, az előírt peremfeltételekre felírt egyenletrendszer megoldásából megkapjuk c_l $(l=1, 2, 3, 4)$ értékeit. Ezután f_k^*-t számítjuk ki és azt u_k-hoz hozzáadva nyerjük a simított f_k értékrendszert, amit szintén az 1. táblázat-

1. TÁBLÁZAT

k	g_k	$\alpha_k^{(1)}$	$\alpha_k^{(2)}$	β_k	u_k	f_k	$f_k + \frac{1}{\varepsilon}\Delta^4 f_{k-2}$
−1	0,78					0,78	
0	0,89					0,89	
1	1,00	0,444 444	−0,11 1111	0,333 333	0,670 807	0,999 061	0,999 997
2	1,11	0,492 308	−0,13 8462	0,645 692	1,071 893	1,108 525	1,110 000
3	1,22	0,491 379	−0,14 0086	0,787 629	1,250 304	1,222 542	1,219 999
4	1,35	0,491 546	−0,14 0149	0,863 986	1,367 407	1,349 688	1,350 000
5	1,49	0,491 706	−0,14 0159	0,940 928	1,493 653	1,490 910	1,489 999
6	1,65	0,491 718	−0,14 0159	1,035 382	1,647 274	1,648 092	1,650 000
7	1,82	0,491 717	−0,14 0159	1,142 504	1,835 540	1,820 386	1,819 999
8	2,01	0,491 717	−0,14 0159	1,261 829	2,073 902	2,012 667	2,010 000
9	2,23	0,491 717	−0,14 0159	1,397 994	2,331 184	2,228 648	2,229 999
10	2,46	0,491 717	0	1,544 934	2,384 504	2,464 042	2,459 999
11	2,72	0	0	1,707 426	1,707 426	2,718 614	2,720 002
12	2,98					2,98	
13	3,24					3,24	

$c_1 = -0{,}000\,020\,796\,0$ $\quad n=11$ $\quad \varrho = 2{,}671\,096\,835$

$c_2 = -0{,}000\,011\,568\,7$ $\quad \varepsilon=3$ $\quad \alpha^\circ = 48{,}950\,406\,73$

$c_3 = 0{,}890\,020\,796\,0$ $\quad s=0{,}279\,751$

$c_4 = 0{,}387\,820\,891\,8$

2. TÁBLÁZAT

	1	2	3	4
x_k	$\Delta^2 g_k$*	$\Delta^2 f_k$	$\Delta^2 e^{\frac{k-1}{10}}$	$g_k - e^{\frac{k-1}{10}}$
$\bar{x}$	0,016 667	0,016 123	0,017 056	0,000 339
σ	0,009 428	0,005 170	0,004 373	0,002 533
$(\overline{x^2})^{1/2}$	0,019 149	0,016 932	0,017 608	0,002 556
	5	6	7	8
x_k	$f_k - e^{\frac{k-1}{10}}$	$\Delta^2 g_k$**	$\Delta^2 u_k$	$u'_k - e^{\frac{k-1}{10}}$
$\bar{x}$	0,000 627	0,011 111	0,017 481	0,001 806
σ	0,002 000	0,073 703	0,004 271	0,047 299
$(\overline{x^2})^{1/2}$	0,002 096	0,074 536	0,017 995	0,047 334

$$\bar{x} = \frac{1}{n}\sum_{k=1}^{n} x_k, \quad \sigma^2 = \frac{\sum_{k=1}^{n} x_k^2}{n} - \left(\frac{\sum_{k=1}^{n} x_k}{n}\right)^2, \quad \overline{x^2} = \frac{1}{n}\sum_{k=1}^{n} x_k^2 = \sigma^2 + \bar{x}^2$$

* $g_k = 2$ tizedesre $e^{\frac{k-1}{10}}$

** $g_k = 1$ tizedesre $e^{\frac{k-1}{10}}$

ban találunk. Az 1. táblázat utolsó oszlopa: $f_k + \frac{1}{\varepsilon}\Delta^4 f_{k-2}$ a számolás ellenőrzésére szolgál, ugyanis ez meg kell egyezzék g_k-val.

A simítás értékeléséhez a simítandó értékrendszer második differenciáinak átlagértékét, szórását és négyzetei átlagának négyzetgyökét a 2. táblázat 1. oszlopában láthatjuk. A 2. oszlopban a simított értékrendszerből, a 3. oszlopban pedig a pontos függvényértékekből számított második differenciák átlagértékét, szórását és négyzetei átlagának négyzetgyökét tüntettük fel. A simítással elértük, hogy a második differenciák szórása csökkent és sokkal jobban megközelíti a pontos második differenciák szórását. A 2. táblázat 4. oszlopában a simítandó és a pontos függvényértékek eltérésének átlagát, szórását és abszolút értékének átlagát, az 5. oszlopban a simított és a pontos függvényértékek különbségeinek hasonló jellemzőit láthatjuk. A simítással csökkent a különbség szórása és a különbség abszolút értékének átlaga.

Második példánkban ugyancsak az e^x függvénynek az előbbi intervallumon felvett értékeinek egy tizedesre kerekített értékeit választottuk simítandó értékrendszerként (3. táblázat). Mivel a g_k számsorozat hibája most nagyobb mint az első példánkban volt, ezért a simítási paraméter értékét most kisebbnek kellett választani: $\varepsilon = 0{,}1$.

A (3.3) peremfeltétel helyett most az intervallum kiterjesztett pontjaiban $e^{\frac{k-1}{10}}$ $(k = 0, -1, 12, 13)$ ugyancsak egy tizedes pontosságú értékét választottuk peremfel-

3. TÁBLÁZAT

k	g_k	$\alpha_k^{(1)}$	$\alpha_k^{(2)}$	β_k	u_k	$u_k+\frac{1}{\varepsilon}\Delta^4 u_{k-2}$	u_k'
−1	0,8	0	0	0,8	0,8		
0	0,9	0	0	0,9	0,9		
1	1,0	0,655 738	−0,16 3934	0,475 410	0,998 534	1,000 074	0,993 412
2	1,1	0,961 810	−0,28 7600	0,319 708	1,100 261	1,099 931	1,049 306
3	1,2	1,117 179	−0,36 7712	0,262 464	1,209 994	1,200 034	1,154 123
4	1,3	1,192 136	−0,41 3531	0,251 306	1,332 513	1,299 993	1,304 443
5	1,5	1,223 165	−0,43 5621	0,266 603	1,471 602	1,500 012	1,473 146
6	1,6	1,233 059	−0,44 4052	0,291 866	1,627 793	1,599 983	1,660 981
7	1,8	1,235 034	−0,44 6353	0,323 098	1,804 459	1,800 059	1,871 375
8	2,0	1,235 086	−0,44 6691	0,358 303	2,002 192	1,999 952	2,083 358
9	2,2	1,235 067	−0,44 6693	0,396 483	2,221 144	2,199 964	2,297 140
10	2,5	1,235 197	−0,44 6737	0,441 349	2,461 243	2,500 023	2,496 193
11	2,7	1,235 351	−0,44 6814	0,488 732	2,720 299	2,699 959	2,692 667
12	3,0	0	0	3,0	3,0		
13	3,3	0	0	3,3	3,3		

$n=11$
$\varepsilon=0{,}1$
$s=0{,}011\ 850$

tételnek. Az inhomogén egyenlet partikuláris megoldását úgy határoztuk meg, hogy az teljesítse ezeket a peremfeltételeket is (így nincs szükség a homogén egyenlet megoldásának külön kiszámolására). Ezt úgy érjük el, hogy $\alpha_k^{(1)}$ és $\alpha_k^{(2)}$ értékét nullának vesszük az eredeti intervallumon kívüli pontokban, továbbá itt $\beta_k=g_k$ $(k=-1, 0, 12, 13)$. Az $\alpha_k^{(1)}$, $\alpha_k^{(2)}$ és β_k értékeit $k=1, 2, \ldots, 11$ esetén a (9.6) képletek alapján, a simított u_k értéket pedig a (9.7) segítségével állítottuk elő. A számolás ellenőrzése végett most is kiszámoltuk $u_k+\frac{1}{\varepsilon}\Delta^4 u_{k-2}$ értékeit, amelyek g_k-val kell megegyezzenek.

A simított u_k értékrendszert numerikusan differenciáltuk a következő formulával:

$$u_k' = \frac{1}{h}\left(\frac{1}{12}u_{k-2}-\frac{2}{3}u_{k-1}+\frac{2}{3}u_{k+1}-\frac{1}{12}u_{k+2}\right), \tag{10.1}$$

ahol $h=x_{k+1}-x_k$ a lépésköz (példánkban 0, 1). A differenciáló formula hibájának nagyságrendje h^4. A közelítő differenciálhányadosok értékét a 3. táblázat utolsó oszlopában láthatjuk.

A 2. táblázat 6. és 7. oszlopában a simítandó és a simított adatrendszer második differenciáinak átlagát, szórását és négyzetei átlagának négyzetgyökét tartalmazza.

A simítás eredményeként elértük azt, hogy a simított u_k számok említett jellemzői nagyon jól megközelítik a 3. oszlopban feltüntetett, a pontos e^x függvény megfelelő értékeit. A közelítő differenciálhányados pontostól való eltérésének a maximuma −5,5%. Végül amint az a 2. táblázat utolsó oszlopa alapján megállapítható, a közelítő differenciálhányados hibája abszolút értékének átlaga 0,047 334, ami kisebb, mint a kiinduló simítandó értékek lehetséges maximális hibája.

Amint a bemutatott példák mutatják, az eljárás $r=2$ választásával rövid, pontos és kényelmes.

* * *

Végezetül megköszönöm KIS OTTÓnak a cikk felépítéséhez adott igen hasznos tanácsait.

IRODALOM

[1] FRANK—MISES, *A mechanika és fizika differenciál- és integrálegyenletei* (Műszaki Kiadó, Budapest, 1967).
[2] GELFOND, A. O., *Differenciaszámítás* (Akadémia Kiadó, Budapest, 1954).
[3] GUEST, P. G, *Numerical Methods of Curve Fitting* (Cambridge at the University Press, 1961).
[4] HAMMING, R. W., *Numerical Methods for Scientists and Engineers* (Mc Graw—Hill, 1973).
[5] MARCSUK, G. I., *A gépi matematika numerikus módszerei* (Műszaki Kiadó, Budapest, 1976).
[6] RALSTON, A., *Bevezetés a numerikus analízisbe* (Műszaki Kiadó, Budapest, 1969).
[7] WENDROFF, P., *Theoretical Numerical Analysis* (Academic Press, New York, 1966).
[8] WHITTAKER, E., *The Calculus of Observations* (Blackie & Son. Ltd., London & Glasgow, 1954).

(Beérkezett: 1980. október 14.)

NYÍRI ANDRÁS
1138 BUDAPEST, RÓBERT KÁROLY KRT. 14/B.

A METHOD FOR SMOOTHING EMPIRICAL FUNCTION

A. NYÍRI

This paper presents a method for smoothing a set of approximate values of a function given at equidistant points. The graduated values obtained minimize both their r-th differences and their deviations from the values to be smoothed simultaneously.

Alkalmazott Matematikai Lapok 6 (1980) 287–290

TÖBBSZÖRÖS VALÓS GYÖKÖKKEL RENDELKEZŐ VALÓS EGYÜTTHATÓS POLINOMOK FAKTORIZÁLÁSA

VARGA GYULA

Budapest

A dolgozatban egy tetszés szerinti ismert multiplicitású valós gyökhöz tartozó gyöktényező és a hányadospolinom egyidejű kiszámítására szolgáló algoritmust ismertetünk. Az eljárás a probléma megoldására másodrendű konvergenciát biztosít.

1. Bevezetés

Polinomok faktorizálására a szakirodalomban számos iterációs eljárás található (l. pl. [1], [3]). Ezek közül a *Newton—Raphson-módszeren alapuló Bairstow-eljárás* egyszeres valós vagy konjugált komplex gyökpárokhoz tartozó másodfokú gyöktényezőket számít ki, és megadja a gyöktényezővel való leosztás hányadospolinomját is. Többszörös valós gyökök esetén a polinom faktorizálása általában úgy történik, hogy a többszörös valós gyök kiszámítása után a gyökhöz tartozó magasabb fokú gyöktényezővel polinomosztást végzünk a hányadospolinom együtthatóinak előállítására. A [2]-ben egy ugyancsak a *Newton—Raphson-módszeren* alapuló eljárást adtunk meg, amely kétszeres valós gyökökhöz tartozó gyöktényezők és a leválasztásuk után adódó hányadospolinom együtthatóinak egyidejű kiszámítását végzi el. Az eljárás alkalmazható valós együtthatós polinomok szélsőérték helyének a meghatározására is.

A jelen dolgozatban általánosítjuk a [2]-ben leírt eljárást tetszésszerinti ismert multiplicitású valós gyökhöz tartozó gyöktényező és a hányadospolinom egyidejű kiszámítására. Az eljárás a probléma megoldására másodrendű konvergenciát biztosít, és egyéb alkalmazásként polinomok szélsőérték-helyei, valamint inflexiós helyei is kiszámíthatók vele.

2. A polinomfaktorizálási algoritmus

Legyen $a(x)$ n-edfokú valós együtthatós polinom, és legyen p egy k-szoros $(0<k<n)$ valós gyökének valamely közelítése. Osszuk el az $a(x)$ polinomot az $(x-p)^k$ tényezővel. Ezt a polinomosztást az alábbi polinomazonosság segítségével írhatjuk fel:

$$a(x) \equiv b(p, x)(x-p)^k + c(p, x), \tag{2.1}$$

ahol $b(p, x)$ a hányadospolinom, a

$$c(p, x) = \sum_{j=0}^{k-1} c_j(p) x^j$$

$k-1$-edfokú polinom pedig az osztási maradék. Az osztás a $(-1)^k\binom{n}{k}p^{n-k}$ együtthatók szukcesszív előállítása után egyszerűen elvégezhető.

Differenciáljuk a (2.1) azonosságot p szerint:

$$0 \equiv -kb(p, x)(x-p)^{k-1}+\frac{\partial b(p, x)}{\partial p}(x-p)^k+\frac{\partial c(p, x)}{\partial p}, \tag{2.2}$$

ahol az osztás maradékpolinomjának p szerinti első deriváltja a

$$\frac{\partial c(p, x)}{\partial p}=\sum_{j=0}^{k-1}\frac{dc_j(p)}{dp}x^j$$

képlettel adható meg. Minthogy

$$\left.\frac{\partial^{j+1}c(p, x)}{\partial p\,\partial^j x}\right|_{x=p}=0 \quad (j=0, 1, \ldots, k-2),$$

ezért szükségképpen fennáll a

$$\frac{\partial c(p, x)}{\partial p}=R(x-p)^{k-1}$$

egyenlőség. A kétféle előállítás főegyütthatóinak összehasonlítása alapján

$$R=\frac{dc_{k-1}(p)}{dp}$$

adódik. Ezt behelyettesítve kapjuk az alábbi azonosságot:

$$0 \equiv -kb(p, x)(x-p)^{k-1}+\frac{\partial b(p, x)}{\partial p}(x-p)^k+\frac{dc_{k-1}(p)}{dp}(x-p)^{k-1}. \tag{2.3}$$

Egyszerűsítve az $(x-p)^{k-1}$ tényezővel, adódik a

$$0 \equiv -kb(p, x)+\frac{\partial b(p, x)}{\partial p}(x-p)+\frac{dc_{k-1}(p)}{dp} \tag{2.4}$$

azonosság. Ezt az $x=p$ helyen véve kapjuk a

$$\frac{dc_{k-1}(p)}{dp}=kb(p, p) \tag{2.5}$$

egyenlőséget.

A továbbiakban keressük meg a $c_{k-1}(p)=0$ egyenlet egy megoldását. Kiindulva egy alkalmas $p^{(0)}$ közelítő értékből, alkalmazhatjuk a *Newton—Raphson-módszert:*

$$p^{(i+1)}=p^{(i)}-\frac{c_{k-1}(p^{(i)})}{c'_{k-1}(p^{(i)})} \quad (i=0, 1, \ldots). \tag{2.6}$$

Az iteráció alkalmazásához szükséges függvényértéket a (2.1) polinomosztás maradékpolinomjának főegyütthatója adja meg, a p szerinti deriváltat pedig a (2.5) egyenlő-

ség. Az iterációs eljárás sikeres befejezése után eltűnik a maradékpolinom főegyütthatója. Emellett érvényes az alábbi tétel:

2.1. TÉTEL. Az alábbi két feltétel teljesülése esetén

1) $\exists p^*: \left.\frac{d^j a(x)}{dx^j}\right|_{x=p^*} = 0 \quad (j = 0, 1, \ldots, k-1),$

2) $|b(p^{(i)}, p^{(i)})| > M > 0 \quad (i = 0, 1, \ldots),$

a (2.6) iterációs eljárás az $a(x)$ polinomnak egy k-szoros gyökéhez konvergál. A konvergencia másodrendű.

Bizonyítás: Írjuk fel a (2.1) azonosságot p^* segítségével:

$$a(x) \equiv b(p^*, x)(x-p^*)^k + c(p^*, x). \tag{2.7}$$

A $c(p^*, x)$ maradékpolinomot az $x=p^*$ körül *Taylor-sorba* fejtve kapjuk az alábbi egyenlőséget:

$$c(p^*, x) = \sum_{j=0}^{k-1} \left.\frac{d^j a(x)}{dx^j}\right|_{x=p^*} \frac{(x-p^*)^j}{j!}. \tag{2.8}$$

Az 1) feltétel alapján látható, hogy $c(p^*, x) \equiv 0$, tehát p^* gyöke az $a(x)$ polinomnak, hiszen érvényes az

$$a(x) = b(p^*, x)(x-p^*)^k \tag{2.9}$$

polinomfelbontás. Mivel továbbá fennáll

$$\left.\frac{\partial^{k-1} c(p^*, x)}{\partial x^{k-1}}\right|_{x=p^*} = \left.\frac{d^{k-1} a(x)}{dx^{k-1}}\right|_{x=p^*} = (k-1)!\, c_{k-1}(p^*) = 0, \tag{2.10}$$

ezért p^* a (2.6) iteráció fixpontja. Végül a 2) feltétel alapján (2.5) figyelembevételével a másodrendű konvergenciát is beláthatjuk, ugyanis

$$c'_{k-1}(p^*) = k!\, b(p^*, p^*) \neq 0.$$

3. Megjegyzések

1) A tétel első feltétele azt jelenti, hogy a keresett gyök multiplicitása ne legyen kisebb k-nál, a második pedig azt, hogy e multiplicitás ne legyen nagyobb k-nál, vagyis a hányadospolinomnak p^* ne legyen gyöke. Ebből következik, hogy mindkét feltétel egyben szükséges is.

2) Mivel

$$\left.\frac{d^{k-1} a(x)}{dx^{k-1}}\right|_{x=p} = (k-1)!\, c_{k-1}(p), \tag{2.11}$$

ezért a tételben foglalt állítás voltaképpen azt fejezi ki, hogy ha egy polinomnak valamely p^* helyen k-szoros gyöke van, akkor a *Newton—Raphson-eljárás* alkalmas kezdő értékből kiindulva másodrendben konvergál a polinom $k-1$ -edik deriváltjának ugyanazon a p^* helyen levő egyszeres gyökéhez.

3) Ha a tétel első feltételében p^*-ra nem kötjük ki az $a(p^*)=0$ egyenlőség teljesülését, akkor a

$$\left.\frac{d^k a(x)}{dx^k}\right|_{x=p^*} = k!\,b(p^*, p^*) \neq 0$$

egyenlőtlenség miatt p^* az $a(x)$ polinomnak páros k esetén szélsőérték-helye, páratlan k esetén pedig inflexiós helye. Az eljárás értelemszerűen ezek kiszámítására is alkalmas.

4) Az eljárás szerint működő program a k-adfokú közelítő gyökténvező együtthatóinak előállítását, a (2.1) polinomosztást, a (2.5) *Horner-eljárást* és a (2.6) iterációt tartalmazza. Bemenő paraméterei a polinom fokszáma és együtthatóinak tömbje, valamint a keresett gyök ismert multiplicitása és kezdő közelítése. Az iteráció sikeres végrehajtása után kimenő paraméterként a p^* k-szoros gyököt, a $b(p^*, x)$ hányadospolinom együtthatóinak tömbjét és (ellenőrzésképpen) az $a(p^*)$ függvényértéket kapjuk meg.

5) Az eljárás FORTRAN szubrutinjának kipróbálása az MTA CDC 3300-as gépén történt.

IRODALOM

[1] RALSTON, A., *Bevezetés a numerikus analízisbe* (Műszaki Könyvkiadó, Budapest, 1969).
[2] VARGA, GY., „Kétszeres valós gyökökkel rendelkező valós együtthatós polinomok faktorizálása", *Alk. Mat. Lapok* 4 (1978) 359—362.
[3] Березин, И. С. и Жидков, Н. П., *Методы вычислений* (Наука, Москва, 1966).

(Beérkezett: 1980. május 8.)

VARGA GYULA
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST, ÚRI U. 49.

FACTORIZATION OF POLYNOMIALS WITH MULTIPLE REAL ROOTS

GY. VARGA

The paper gives a procedure based on the *Newton-Raphson method* for factorization of real polynomials which have multiple real roots of known multiplicity. The convergence of the procedure is quadratic.

Alkalmazott Matematikai Lapok 6 (1980) 291–311

POPULÁCIÓS MODELLEK ÉS KISZOLGÁLÓ HÁLÓZATOK

GÁRDOS ÉVA ÉS TÖRÖK TURUL

Budapest

A populációs modellek az alkalmazott matematika számos területén hatékonynak bizonyulnak. Ezek a modellek egymással kapcsolatban álló helyek között vándorló egyedek, vagy diszkrét anyagmennyiségek időbeli eloszlását véve alapul, a komplex viselkedést írják le.

A számítógép-hálózatok elméletében is jól alkalmazható eredményeket igyekeztünk összefoglalni, lehetőleg úgy csoportosítva őket, hogy az általánosítási lehetőségek, illetve azok akadályai kitűnjenek. Ezenkívül felsoroltunk néhány más kísérletet is, melyek többnyire nem teljesen kidolgozottak, de talán akad közöttük további vizsgálatra érdemes elmélet is.

1. Bevezetés

Több tudományterületen találkozhatunk problémákkal, amelyeknek matematikai tárgyalása hasonló. Populációs modelleket használnak a reakciókinetikában, számítógépek tervezésénél, közlekedési és szállítási hálózatoknál, üzleti életben, különböző nyilvántartások készítésénél, stb. Ennek megfelelően bizonyos időközönként megnő a témakörben közzétett publikációk száma. Egy ilyen hullámhegy tart a hetvenes évek közepétől napjainkig, a cikkek szinte követhetetlen áradatával sokkolva a kutatókat. Ez a tény feltétlenül indokol egy rövid áttekintést. Másrészt néhány további — még nem teljesen kidolgozott — lehetőséget is számba veszünk.

A témakör feladata verbálisan a következőképpen fogalmazható meg:

(1.1)

VALAKIKKEL | VALAHOL | VALAHOGYAN | VALAMI TÖRTÉNIK.

A véletlenszerű történésekből adódó konfliktushelyzetek kiértékelését végezzük el, elsősorban a különböző helyek népességének a vizsgálatával.

A populációs modell elnevezéssel szinonim kifejezések a következők (zárójelben az angol terminológia):

kiszolgáló hálózatok *(service networks)*
sorbanállási hálózatok *(queueing networks)*
számítógép-hálózatok *(computer networks)*
vándorlási modellek *(migration models)*
több dimenziós születési-kihalási folyamat *(birth-death vector process)*
rekesz rendszerek *(compartment, cell models) stb.*

Jóllehet a hetvenes évek említett hullámhegyét elsősorban a számítástechnika élteti, mégsem kizárólag számítógéphálózatok tárgyalását tűztük célul, ennél jóval általánosabban igyekszünk fogalmazni.

A részletes tárgyalás a következő lépésekben történik. A második pontban egzaktabban, és több oldalról próbáljuk körüljárni a feladatot, és eljutunk a *Markov populációs folyamatok* definíciójához.

Ezeknek az igen nagy állapotterű *Markov folyamatoknak* a határeloszlásához vezető utakat jelöli ki a 3. pont. Az itt vázolt módszerek részletes kifejtését és alkalmazását adja a 4. pont. Az ismertebb eredmények összefoglalása itt történik.

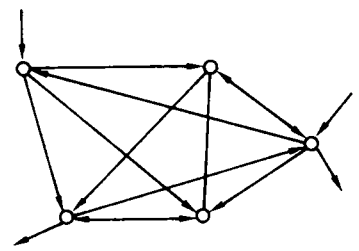
Az ötödik pontban a legújabb — és talán kevésbé elterjedt modelleket foglaljuk össze. Ezek a klasszikus modellek közvetlen általánosításai — persze kicsit más, bonyolultabb eszközökkel.

Végül, a hatodik részben megvizsgálunk néhány, egészen más eszközökkel operáló lehetőséget is. Ezek nem dicsekedhetnek az előzőkhöz hasonló szép hagyományokkal, sem végleges leírással, hiszen csak a legutóbbi időkben fogalmazódtak meg.

2. Matematikai megfogalmazás

Tömegkiszolgáló rendszereket (M darab) kapcsolunk össze, oly módon, hogy bizonyosok kimenő folyamatai mások bemenő folyamatába torkollanak. Kérdés az — esetleg nem független — állomások együttes viselkedése.

A vizsgálat tárgya egy M szögpontú, irányított gráf. A szögpontok sorbanállási rendszerek, melyek közt az igények (az élek mentén) vándorolhatnak. Élek mutathatnak a gráfba kívülről és a gráfból kívülre is.



1. ábra

Legyen $\mathcal{N}_M = N^+ \times N^+ \times \ldots \times N^+$ a nemnegatív egész komponensű, M dimenziós vektorok tere. Definiáljuk $\xi(t)$ sztochasztikus folyamatot $\mathcal{N}_M$ állapottérrel, mely leírja a hálózat csomópontjaiban tartózkodók számának változását az idő függvényében:

$\{\xi(t)=\mathbf{n}\}$ az az esemény, hogy t időpontban $\{\xi_i(t)=n_i;\ 1 \leqq i \leqq M\}$.

$\xi(t)$-t *populációs folyamatnak* nevezzük. Teljes általánosságban teljességgel kezelhetetlen, ezért további feltételekkel kell élnünk. Megkülönböztetett jelentőséget tulajdonítunk az alábbi típusú változásoknak:

$$\{\xi(t) = \mathbf{n} \to \xi(t+\Delta t) = \mathbf{n}+\mathbf{e}_j\} \qquad \text{„beérkezés”}$$

$$\{\xi(t) = \mathbf{n} \to \xi(t+\Delta t) = \mathbf{n}-\mathbf{e}_i\} \qquad \text{„távozás”}$$

$$\{\xi(t) = \mathbf{n} \to \xi(t+\Delta t) = \mathbf{n}-\mathbf{e}_i+\mathbf{e}_j\} \qquad \text{„átmenet”}$$

itt $\mathbf{e}_k$ a k-adik egységvektort jelöli: minden komponense nulla, kivéve a k-adik, amely eggyel egyenlő.

Ha feltesszük, hogy rövid idő alatt (Δt) ezek a változások Δt-vel egyenesen arányos valószínűséggel következnek be, más változás valószínűsége pedig $o(\Delta t)$ nagyságrendű, akkor *Markov populációs folyamattal* van dolgunk.

Ekkor tehát definiálhatjuk a folyamat infinitezimális átmenet-sűrűségeit:

$$
\begin{aligned}
&Q\{\xi(t)\colon \mathbf{n} \to \mathbf{n}+\mathbf{e}_j\} = \alpha_j(\mathbf{n}) \\
&Q\{\xi(t)\colon \mathbf{n} \to \mathbf{n}-\mathbf{e}_i\} = \beta_i(\mathbf{n}) \\
&Q\{\xi(t)\colon \mathbf{n} \to \mathbf{n}+\mathbf{e}_j-\mathbf{e}_i\} = \gamma_{ij}(\mathbf{n})
\end{aligned}
\tag{2.1}
$$

Igaz, hogy az így definiált modellek kezelése is eléggé nehézkes, további megszorítások szükségesek, de legalább elméletileg megoldhatóak. Ez azt jelenti, hogy kereshetjük stacionárius eloszlásukat, és más kérdésekre is választ kaphatunk. Az eddig tartó valószínűségszámítási elmélet helyét a lineáris algebra és általában a numerikus módszerek vehetik át.

Érdemes felfigyelni arra a kitételre, hogy a változások valószínűségei arányosak az eltelt idővel. Ez azt jelenti, hogy a hálózatban a „történések" exponenciális eloszlású idők alatt mennek végbe, vagyis csupa M/M/1 kiszolgáló rendszert kapcsoltunk össze, és a hálózatba az igények *Poisson-folyamatként* lépnek be. Megjegyezzük, hogy a diszkrét időeloszlások mellett definiált populációs folyamat csak akkor lesz markovi, ha az idők geometriai eloszlásúak, azaz minden időegységben megmondjuk, mekkora valószínűséggel következik be egy esemény (függetlenül attól, mikor következett be utoljára). M/D/1 kiszolgáló rendszerek összekapcsolásából általában nem kapunk *Markov populációs folyamatot.*

3. Aszimptotikus viselkedés. Egyensúlyi feltételek

Markov-láncok vizsgálatánál alapvető jelentőségű a stacionárius eloszlás meghatározása. Ez azt jelenti, hogy keressük a $P\{\xi(t)=\mathbf{n}\}$ valószínűségek határértékét, amidőn $t \to \infty$. Az eloszlás létezését feltesszük (valóban legtöbbször létezik, erre még a következő fejezetben visszatérünk) és a *Kolmogorov-egyenletek*

$$
\begin{aligned}
P(\mathbf{n})\Big[\sum_i \alpha_i(\mathbf{n}) + \sum_i \beta_i(\mathbf{n}) + \sum_i \sum_j \gamma_{ij}(\mathbf{n})\Big] = \\
= \sum_i P(\mathbf{n}-\mathbf{e}_i)\alpha_i(\mathbf{n}-\mathbf{e}_i) + \sum_i P(\mathbf{n}+\mathbf{e}_i)\beta_i(\mathbf{n}+\mathbf{e}_i) + \\
+ \sum_i \sum_j P(\mathbf{n}-\mathbf{e}_i+\mathbf{e}_j)\gamma_{ji}(\mathbf{n}-\mathbf{e}_i+\mathbf{e}_j)
\end{aligned}
\tag{3.1}
$$

megoldására lehet koncentrálni, vagy ami az állapottér átszámozása után evvel ekvivalens, az

$$
\mathbf{x} = \mathbf{x}\mathbf{P} \tag{3.2}
$$

fixpont keresésére. ($\mathbf{P}$ itt már sztochasztikus mátrix.)

Ezzel a feladat még korántsem tekinthető megoldottnak, hiszen (3.1)-ben az ismeretlenek (az egyes állapotok $P(\mathbf{n})$ valószínűségei), és így az egyenletek száma ∞ is lehet, és egy végtelen nagy átmenet intenzitás mátrix kitöltése szükséges.

Kedvező esetben sikerülhet egy ügyes korlátozással (pl. jelen levő igények száma) az állapotteret végessé alakítani. Ez a véges tér még mindig igen nagy lehet. Például 5 csomópont és max 10 igény ($|\mathbf{n}| = \sum n_i \leqq 10$) esetén az állapottér 3003 elemű, tehát az átmenetek száma közel 10^7. Igaz, hogy a mátrix igen ritka, kitöltöttsége 1% alatt van, de teljes általánosságban az elemek felírása szinte reménytelen feladat. Reménytelen például az adatok mérése szempontjából, s emellett a tárolási gondok már eltörpülnek.

Röviden ismertetünk egy igen hasznos tulajdonságot, amely jelentősen megkönnyítheti a stacionárius eloszlás meghatározását (reverzibilitás). (3.2) kis átalakítással a következőképpen írható

$$\mathbf{x}(\mathbf{P} - \operatorname{diag} \mathbf{P}) = \mathbf{x}\mathbf{D},$$

ahol $\mathbf{D}$ egy diagonális mátrix, i-edik sorában $\sum_{\substack{j=1 \\ i \neq j}}^{\infty} p_{ij}$ áll. Ezt soronként kiírva

$$\sum_{\substack{j=1 \\ i \neq j}}^{\infty} x_j p_{ji} = \sum_{\substack{j=1 \\ i \neq j}}^{\infty} x_i p_{ij} \tag{3.3}$$

jutunk a *Kolmogorov-egyenletekhez,* amit globális egyensúlynak is neveznek. Ennek megoldása például *Gauss-eliminációval* $O(N^3)$ műveletigényű, ahol N az állapottér (S) számossága.

Ha feltesszük, hogy (3.3) helyett a sokkal szigorúbb

$$x_j p_{ji} = x_i p_{ij}, \quad i, j \in S \tag{3.4}$$

feltételek teljesülnek, a műveletigény $O(N)$-re csökken. (3.4)-t részletes egyensúlyi feltételnek nevezik. Azokat a *Markov-láncokat,* amelyekhez létezik $\mathbf{x}$ vektor, hogy (3.4) teljesül, reverzibilisnek nevezik. Könnyű belátni, hogy ilyenkor $\mathbf{x}$ éppen arányos a stacionárius eloszlással, kielégíti (3.2)-t. Ebben rejlik a hátránya is: a reverzibilitás teszteléséhez a stacionárius eloszlás ismerete szükséges, és ekkor már semmi haszna a kisebb műveletigénynek. Gyakorlatilag járhatóbb út, hogy feltételezzük a reverzibilitást, és (3.4) alapján rekurzív módon meghatározzuk $\mathbf{x}$ vektort — $x_1 = 1$ kiindulással. Ha közben nem jutunk ellentmondásra (hiszen $n(n-1)/2$ egyenletünk van n helyett), akkor a lánc reverzibilis, és $\mathbf{x}$ — normálás után — a stacionárius eloszlás. Ha a részletes egyensúlyi feltételek között van ellentmondás, akkor a lánc nem reverzibilis, és $\mathbf{x}$ csak kezdeti eloszlásként jöhet szóba.

3.1. TÉTEL [24]. A reverzibilitás teljesüléséhez szükséges feltétel, hogy az átmenetvalószínűség mátrix előjel-szimmetrikus legyen, azaz

$$p_{ij} = 0 \Leftrightarrow p_{ji} = 0.$$

3.2. TÉTEL [24]. Egy *Markov-lánc* akkor és csakis akkor reverzibilis, ha tetszőlegesen választott $i_1, i_2, \ldots, i_k \in S$ állapotokra

$$p_{i_1 i_2} \cdot p_{i_2 i_3} \cdot \ldots \cdot p_{i_{k-1} i_k} p_{i_k i_1} = p_{i_1 i_k} \cdot p_{i_k i_{k-1}} \cdot \ldots \cdot p_{i_3 i_1} p_{i_2 i_1}. \tag{3.5}$$

(3.5)-t *Kolmogorov-féle ciklusfeltételnek*, nevezik. Könnyen belátható, hogy (3.5) helyett elegendő

$$p_{ij} \cdot p_{jk} \cdot p_{ki} = p_{ik} \cdot p_{kj} \cdot p_{ji}$$

három hosszúságú ciklus feltételek teljesülése a reverzibilitáshoz, tetszőleges $i, j, k \in S$ választással.

Többször használjuk majd a „szorzatforma" fogalmát. $P(\mathbf{n})$ stacionárius eloszlást szorzat formájúnak nevezzük, ha

$$P(\mathbf{n}) = C \cdot g(\mathbf{n}) \cdot \sum_{i=1}^{M} p_i(n_i)$$

alakú, ahol C egy normáló konstans, mely a

$$\sum_{\mathbf{n} \in \mathcal{X}_M} P(\mathbf{n}) = 1$$

feltételt hivatott szavatolni.

A továbbiakban részletes, valamint (3.3) és (3.4) „közé eső" egyensúlyi feltételeket keresünk, és ezeket alkalmazzuk (3.1) megoldásához. A kapott eredmények mind szorzat formájúak lesznek.

4. Klasszikus populációs modellek

Írjuk fel a reverzibilitási feltételeket a (3.1) rendszerhez

$$P(\mathbf{n})\alpha_i(\mathbf{n}) = P(\mathbf{n}+\mathbf{e}_i)\beta_i(\mathbf{n}+\mathbf{e}_i)$$

$$P(\mathbf{n})\beta_i(\mathbf{n}) = P(\mathbf{n}-\mathbf{e}_i)\alpha_i(\mathbf{n}-\mathbf{e}_i)$$

$$P(\mathbf{n})\gamma_{ij}(\mathbf{n}) = P(\mathbf{n}-\mathbf{e}_i+\mathbf{e}_j)\gamma_{ji}(\mathbf{n}).$$

Ennek tesztelése igen körülményes, ezért további feltételek mellett keresünk egyszerűbb lehetőségeket.

4.1. TÉTEL [21]. Ha $\alpha_i = \beta_i = 0$ $i \in \{1, \ldots, M\}$ esetén, és

$$\gamma_{ij}(\mathbf{n}) = p_{ij}\Phi_i(n_i)\Psi_j(n_j)f(\mathbf{n}) \tag{4.1}$$

alakú, akkor a $\mathbf{P} = [p_{ij}]$ irányítás mátrix által meghatározott $\{1, 2, \ldots, M\}$ állapotterű *Markov-lánc*, és $\xi(t)$ *Markov populációs folyamat* egyszerre lesz reverzibilis. Továbbá

$$P(\mathbf{n}) = (c/f(\mathbf{n})) \sum_{i=1}^{M} x_i^{n_i} \prod_{r=1}^{n_i} (\Phi_i(r))^{-1}, \tag{4.2}$$

ahol

$$x_i p_{ij} = x_j p_{ji}. \tag{4.3}$$

Ennek egy speciális esete [7] III. modellje:

$$\gamma_{ij}(\mathbf{n}) = g_{ij} \cdot d_i \cdot n_i \cdot (a_j + b_j n_j).$$

4.2. TÉTEL [21] Ha $\gamma_{ij}(\mathbf{n}) = 0$ és

$$\alpha_i(\mathbf{n}) = \alpha_i(n_i), \quad \beta_i(\mathbf{n}) = \beta_i(n_i) \tag{4.4}$$

akkor $\xi(t)$ reverzibilis és $P(\mathbf{n})$ szorzat alakú

$$P(\mathbf{n}) = C \cdot \sum_{i=1}^{M} \prod_{r=1}^{n_i} \frac{\alpha_i(r)}{\beta_i(r)}.$$

Speciális eset [7] II. modellje:

$$\alpha_i(n_i) = a_i + b_i n_i, \quad \beta_i(n_i) = d_i \cdot n_i.$$

4.3. TÉTEL. Ha az irányítás reverzibilis (4.3), és (4.4) mellett

$$\frac{\alpha_i(n_i) \cdot \Phi_i(n_i+1)}{x_i \cdot \beta_i(n_i+1) \cdot \Psi_i(n_i)} = A$$

nem függ sem n_i-től, sem i-től, akkor $\xi(t)$ is revezíbilis, és $P(\mathbf{n})$ szorzat alakú, nevezetesen érvényes (4.2), de x_i helyett $A \cdot x_i$-vel.

(3.1) és (3.4) feltételek két végletet jelentenek: a globális és a teljesen részletes egyensúlyt. Heurisztikusan közéjük kívánkozna valami „kevésbé részletes" feltétel. $\alpha_i = \beta_i = 0$ esetben megvizsgáljuk a

$$P(\mathbf{n}) \sum_{i=1}^{M} \gamma_{ij}(\mathbf{n}) = \sum_{i=1}^{M} P(\mathbf{n}+\mathbf{e}_j-\mathbf{e}_i)\gamma_{ji}(\mathbf{n}+\mathbf{e}_j-\mathbf{e}_i) \tag{4.5}$$

$$P(\mathbf{n}) \sum_{j=1}^{M} \gamma_{ij}(\mathbf{n}) = \sum_{j=1}^{M} P(\mathbf{n}+\mathbf{e}_j-\mathbf{e}_i)\gamma_{ji}(\mathbf{n}+\mathbf{e}_j-\mathbf{e}_i) \tag{4.6}$$

feltételeket. $\alpha_i = \beta_i = 0$ azt jelenti, hogy a hálózat zárt, se ki-, se beáramlás nincs, egy rögzített populáció oszlik el a csomópontok között. Jelölje K ezt a rögzített létszámot, és ekkor (M, K) jellemzi a zárt modellt. $A(M, K)$-val jelölve egy ilyen modell állapotainak a számát, ismert, hogy $A(M, K) = \binom{M+K-1}{M-1}$, ami igen nagy lehet. Például $A(5, 10) = 1001$, vagyis a stacionárius eloszlás műveletigénye 10^9!

4.4. TÉTEL. Ha (4.1) teljesül, akkor (4.5) és a reverzibilitás ekvivalensek.

Bizonyítás. Csak (4.5)⇒(3.4) a feladat, fordítva triviális.

Minthogy a gondolatmenet elég bonyolult, párhuzamosan egy példával $(M, K)) = (3, 4)$ modell) illusztráljuk.

Induljunk ki $\mathbf{n} = K \cdot \mathbf{e}_1$ állapotból, és legyen $P(K \cdot \mathbf{e}_1) = 1$. (Esetünkben $P(4, 0, 0) = 1$.) (4.5) egyenletei között van $M-1$ olyan, amelynek a bal oldalán éppen $P(K \cdot \mathbf{e}_1)$ áll. Ezek mind egytagúak, és így egyismeretlenesek. Ily módon kifejezhetjük az összes $(K-1)\mathbf{e}_1 + \mathbf{e}_r$ alakú állapot valószínűségét. (Példánkban $P(3, 1, 0) = \gamma_{12}(4, 0, 0)/\gamma_{21}(3, 1, 0)$ és $P(3, 0, 1) = \gamma_{12}(4, 0, 0)/\gamma_{31}(3, 0, 1)$.) (4.5) egyenletei közül azok, melyek bal oldalán ilyen állapot áll, egy vagy több tagúak. Előbbiek egyismeretlenesek lévén vagy már szerepeltek, (például $P(3, 0, 1) \cdot \gamma_{31}(3, 0, 1) = P(4, 0, 0) \cdot \gamma_{13}(4, 0, 0)$), vagy újabb ismeretlen fejezhető ki belőlük $\left(\text{például } P(2, 2, 0) = \frac{\gamma_{12}(4, 0, 0)}{\gamma_{21}(3, 1, 0)} \cdot \frac{\gamma_{12}(3, 1, 0)}{\gamma_{21}(2, 2, 0)}\right)$. Könnyen látható, hogy a több tagúak legfeljebb egy ismeretlent tartalmazhatnak. Ha valóban van ismeretlen, kifejezzük. $(P\{2, 1, 1) = \{P(3, 1, 0)[\gamma_{13}(3, 1, 0) + \gamma_{23}(3, 1, 0)] - P(3, 0, 1) \cdot \gamma_{32}(3, 0, 1)\} \cdot \{\gamma_{31}(2, 1, 1)\}^{-1})$ Az újonnan kapott állapotokkal a bal olda-

lon ismét vesszük (4.5) egyenleteit, és folytatjuk az eljárást amíg valamennyi egyenlet sorra nem került. Ennek során az ismeretlent még tartalmazó egyenletek nem okoznak gondot. (Például $P(1, 2, 1)=\{P(2, 1, 1)[\gamma_{12}+\gamma_{32}]-P(2, 2, 0)\cdot\gamma_{23}\}\cdot\{\gamma_{21}(1, 2, 1)\}^{-1}$). Ha nincs ismeretlen, akkor az ellentmondás-mentességet kell belátni, azaz, hogy az eddigi értékek valóban egyenlőséget adnak. Könnyen látható, hogy a behelyettesítés után $\varphi_i(n_i)$, $\psi_j(n_j)$, $f(\mathbf{n})$ tényezők minden tagban ugyanazok, csak p_{ij}-kből álló tagok maradnak. Ezekből pedig átrendezés után pontosan a *Kolmogorov-féle ciklusfeltételek* vezethetők le. (Esetünkben például a

$$P(2, 1, 1)\cdot[\gamma_{21}+\gamma_{31}] = P(3, 0, 1)\cdot\gamma_{12}+P(3, 1, 0)\cdot\gamma_{13}$$

egyenletből

$$\frac{p_{12}\cdot p_{23}}{p_{31}}+\frac{p_{12}\cdot p_{23}}{p_{21}}-\frac{p_{13}\cdot p_{32}\cdot p_{21}}{p_{31}\cdot p_{31}}-\frac{p_{13}\cdot p_{32}}{p_{31}}=0$$

adódik, és átrendezéssel kapjuk, hogy

$$\frac{p_{31}+p_{21}}{p_{31}p_{31}\cdot p_{21}}(p_{12}p_{23}p_{31}-p_{13}p_{32}p_{31})=0.$$

Ebből az irányítás reverzíbilis volta és a 4.3. tétel alapján $\xi(t)$ reverzibilitása következik.

Megjegyzések. 1. A (4.1) feltétel nélkül (3.4) és (4.5) ekvivalenciája nem igaz. Könnyű (3, 2) modellt konstruálni, hogy (4.5) nem vonja maga után (3.4)-t. Az állapotokat (2, 0, 0), (1, 1, 0), (1, 0, 1), (0, 2, 0), (0, 1, 1), (0, 0, 2) módon sorba rendezve, és 1—6-ig átjelölve $q_{13}=q_{21}=q_{23}=q_{24}=q_{32}=q_{42}=q_{45}=q_{54}=q_{56}=q_{63}=q_{65}=1$; $q_{31}=$ $=q_{12}=2$; $q=4$; $q_{35}=3$; $q_{52}=\frac{3}{4}$ átmenetintenzitásokkal $\mathbf{Q}$ mátrix nem előjel-szimmetrikus, tehát $\xi(t)$ nem reverzíbilis, de (4.5) minden egyenletét kielégíti a stacionárius eloszlás: $p_1=\frac{2}{19}$, $p_2=p_4=p_5=p_6=\frac{4}{19}$, $p_3=\frac{2}{19}$.

2. Jól ellenőrizhető kritérium hiányában (4.5) teljesülését általában úgy a legegyszerűbb vizsgálni, hogy elvégezzük a bizonyításban vázolt egyszerű rekurziót, és ha nem vezet ellentmondásra, akkor szerencsénk van. Ennek az eljárásnak a műveletigénye egyébként $O(K\cdot A(M, K))$.

4.5. TÉTEL.

(4.7) $$\gamma_{ij}(\mathbf{n}) = p_{ij}\Phi_i(n_i)f(\mathbf{n}),$$

akkor (4.6) automatikusan teljesül. (4.6) egyenleteinek a száma legfeljebb $A(M, K-1)\cdot M$, és megoldásának műveletigénye $A(M, K-1)\cdot O(M^3)$. Ekkor

(4.8) $$P(\mathbf{n}) = \frac{C}{f(\mathbf{n})}\prod_{i=1}^{M} x_i^{n_i}\prod_{r=1}^{n_i}\frac{1}{\Phi_i(r)},$$

ahol

(4.9) $$\mathbf{x} = \mathbf{xP}.$$

Bizonyítás. Az $(M, K-1)$ modell állapotait sorba-rendezve, minden m_i állapothoz definiálhatjuk (M, K) modell állapotának egy C_i csoportját:

$$C_i = \{\mathbf{m}_i + \mathbf{e}_j | j = 1, \ldots, M\}.$$

Ezáltal $A(M, K-1)$ csoporthoz jutunk, mindegyik pontosan M elemet tartalmaz. Természetesen a csoportok nem diszjunktak, minden $\mathbf{n} \in (M, K)$ állapot pontosan $\sum_{i=1}^{M} \operatorname{sign} n_i$ csoportban szerepel. Pl. $M=K=3$ esetben a csoportok

$$\begin{array}{cccccc} (2,0,0) & (1,1,0) & (1,0,1) & (0,2,0) & (0,1,1) & (0,0,2) \in (M, K-1) \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ (3,0,0) & (2,1,0) & (2,0,1) & (1,2,0) & (1,1,1) & (1,0,2) \\ (2,1,0) & (1,2,0) & (1,1,1) & (0,3,0) & (0,2,1) & (0,1,2) \\ (2,0,1) & (1,1,1) & (1,0,2) & (0,2,1) & (0,1,2) & (0,0,3) \end{array}$$

A (4.6) egyenleteket ugyanígy csoportokba sorolva: E_k-ba azok az egyenletek kerülnek, amelyek bal, illetve jobb oldalán $\mathbf{n} \in C_k$, $\mathbf{n} + \mathbf{e}_j - \mathbf{e}_i \in C_k$. Könnyű belátni, hogy minden E_i M darab (homogén) egyenletet tartalmaz, és ezek M ismeretlenre (C elemeire) vonatkoznak. Összesen van $A(M, K-1)$. egyenletünk, és $A(M, K)$ ismeretlenünk. A $K=1$ triviális esettől eltekintve ez nem biztos, hogy megoldható, de ha igen, a műveletigény adódik. Kérdés, mikor oldhatók meg az egyenletek?

(4.6)-ba beírva (4.7)-et és $y(\mathbf{n}) = p(\mathbf{n}) \cdot f(\mathbf{n})$ helyettesítéssel élve

$$y(\mathbf{n}) \cdot \Phi_i(n_i) \sum_{j=1}^{M} p_{ij} = \sum_{j=1}^{M} y(\mathbf{n} + \mathbf{e}_j - \mathbf{e}_i) \cdot \Phi_j(n_j + 1) \cdot p_{ij}$$

adódik. Ezek az egyenletek minden i-re azon E_k-ba tartoznak, amelyet $(\mathbf{n} - \mathbf{e}_i) \to C_k$ megfeleltetéssel definiáltunk. Ezután a tétel verifikálható, ha (4.8)-at behelyettesítjük.

Megjegyzések 1. A tételből az alábbi ismert modellek következnek:

[15] $\Phi_i(n_i) = \mu_i \min(n_i, r_i)$, ahol r_i és μ_i tetszőleges konstansok; $f(\mathbf{n}) = 1$,

[3] $f(\mathbf{n}) = 1$,

[30] $f(\mathbf{n}) = 1$.

A szerzők úgy bizonyítják be eredményeiket, hogy egyszerűen behelyettesítik a megsejtett határeloszlásokat (3.1)-be. Ezáltal elsikkad a részletesebb egyensúly lényege, vagyis az, hogy (4.7) mellett (3.1) és (4.5) ekvivalens, sőt az összehasonlíthatatlanul egyszerűbb (4.9)-re redukálódik.

2. A tételben lényeges, hogy (4.7)-ből hiányzik a $\psi_j(n_j)$ tényező! Könnyű erre ellenpéldát konstruálni (3,2) zárt modellből:

$$p_{13} = p_{32} = p_{21} = 1; \quad \Phi_i(r_i) = 1; \quad \Psi_i(1) = 1, \quad \Psi_i(2) = \frac{1}{2}$$

esetén (3.1)-ből kapott stacionárius eloszlás $\left(\frac{1}{9}, \frac{2}{9}, \frac{2}{9}, \frac{1}{9}, \frac{2}{9}, \frac{1}{9}\right)$ nem elégíti ki (4.6)-ot.

3. A (4.6) egyenleteket több szerző lokális egyensúlyi feltételnek nevezi, utalva arra, hogy ilyenkor nemcsak az összes állapotváltozás eredői vannak egyensúlyban, hanem azok is amelyek csak egy-egy csomópontot érintenek.

4.6. TÉTEL. Ha

$$\alpha_i(\mathbf{n}) = \alpha_i a(|\mathbf{n}|) f(\mathbf{n})$$

$$\beta_i(\mathbf{n}) = \beta_i \Phi_i(n_i) f(\mathbf{n})$$

$$\gamma_{ij}(\mathbf{n}) = p_{ij} \Phi_i(n_i) f(\mathbf{n})$$

akkor $\beta_i + \sum p_{ij} = 1$ mellett teljesülnek

$$P(\mathbf{n}) \sum_{i=1}^{M} \alpha_i(\mathbf{n}) = \sum_{i=1}^{M} P(\mathbf{n}+\mathbf{e}_i)\beta_i(\mathbf{n}+\mathbf{e}_i)$$

$$P(\mathbf{n})\left[\beta_i(\mathbf{n}) + \sum_j \gamma_{ij}(\mathbf{n})\right] = P(\mathbf{n}-\mathbf{e}_i)\alpha_i(\mathbf{n}-\mathbf{e}_i) + \sum_j \gamma_{ji}(\mathbf{n}-\mathbf{e}_i+\mathbf{e}_j) P(\mathbf{n}-\mathbf{e}_i+\mathbf{e}_j)$$

feltételek, és ezek

$$x_i = \alpha_i + \sum_j x_j p_{ij}$$

rendszerre redukálódnak.

$$P(\mathbf{n}) = \frac{C}{f(\mathbf{n})} \prod_{k=1}^{|\mathbf{n}|} a(k) \prod_{i=1}^{M} x_i^{n_i} \prod_{r=1}^{n_i} \frac{1}{\Phi_i(r)}. \tag{4.10}$$

Bizonyítás. A tétel az előző általánosítása nyílt rendszerre, és a bizonyítás is teljesen hasonlóan végezhető el.

Megjegyzések. 1. Speciális esetek:

[16] $a(k) = 1, \quad f(\mathbf{n}) = 1$

[17] $f(\mathbf{n}) = 1$

[12] $a(k) = 1, \quad \Phi_i(r_i) = n_i, \quad f(\mathbf{n}) = 1$

[31] $a(k) = 1, \quad f(\mathbf{n}) = 1$

2. Említettük, hogy egy stacionárius eloszlást szorzat formájúnak nevezzük, ha

$$P(\mathbf{n}) = C \cdot g(\mathbf{n}) \cdot \sum_{i=1}^{M} p_i(n_i)$$

alakú. Valamennyi eredmény ebbe a kategóriába tartozik, és szerepelt példa, hogy a szorzat forma nem teljesült. Sikerült tehát a lokális egyensúly és a szorzatforma kapcsolatát igazolni néhány esetben.

Még egyszer nyomatékosan fel kívánjuk hívni a figyelmet, miszerint jelen fejezet tárgya és lényege a különböző egyensúlyok hierarchiába állítása. (3.1) golbális egyensúly minden ergodikus *Markov-láncra* fennáll. (4.0) részletes egyensúly (reverzibilitás) egy igen erős megszorítás, ritkán teljesül. Kettőjük között van (4.5) és (4.6). Előbbi „majdnem" ekvivalens a reverzibilitással, miként 4.4 tétel és annak 1. megjegyzése tanúsítja. (4.6) — és (4.10) is — már jóval általánosabb feltételek, a populációs

modellek „igazi" eredményei bennük gyökereznek (lokális, vagy csomópontonkénti egyensúly).

Hangsúlyozni kellett ezt, mert például KLEINROCK könyvének is ezek a leggyengébb pontjai: a különböző egyensúlyok eléggé keverednek. KELLY most megjelent munkája ([20]) bizonyára segít a pontos megértésben.

Van még egy adósságunk, annak a vizsgálata, mikor létezik stacionárius eloszlás. Mint ismeretes, ehhez a lánc irreducibilis és aperiodikus volta kell, valamint az, hogy az állapotok pozitív visszatérők legyenek.

Zárt modellben az előbbiekhez lényegében elégséges, hogy az irányítási mátrix irreducibilis és aperiodikus legyen, és $\varphi_i(n_i)=0$ csak $n_i=0$ esetben álljon fenn. Ha $\psi_i(n_i)=0$ megengedett bizonyos állapotokra, akkor ezek tranziensek lesznek. Ugyanígy $f(\mathbf{n})=0$ elnyelő állapotokat generál.

Nyílt esetben a fentiekhez kívánkozik még az $a(k)=0$ felülvizsgálata $\psi_i(n_i)$-hoz hasonló értelemben.

Ezek alapján az irreducibilitás (és aperiodicitás) elég könnyen eldönthető. Ha ez pozitív eredménnnyel jár, érdemes az egyenletrendszer megoldásaként keresni a stacionárius eloszlást. A kérdés már csak az, hogy valódi eloszlást kapunk-e, vagyis teljesül-e a normálási feltétel:

$$\sum_{\mathbf{n}\in S} P(\mathbf{n}) = 1.$$

Ezt hivatott biztosítani a mindenütt előforduló C, az úgynevezett normáló konstans. Ha $C<\infty$, akkor normálható, és a lánc pozitív visszatérő. Zárt esetben mindig ez a helyzet, lévén az állapottér véges. Nyílt rendszernél is lehet véges az állapottér, ha létezik egy K egész szám, hogy $a(r)=0$ minden $r>K$ esetén. Különben elégséges feltétel például, hogy létezik A, melyre

$$\frac{\max_l \{a(l)\} \cdot x_i \cdot \psi_i(r)}{\varphi_i(r)} < q < 1$$

minden $r>K$ és tetszőleges i-re.

Felmerül a kérdés, van-e lehetőség úgy módosítani a klasszikus modellek feltételeit, hogy az exponenciális eloszlás helyett általánosabb eloszlású kiszolgálási idő mellett is szorzat alakú legyen a rendszer stacionárius eloszlása.

Amennyiben az egyes csomópontokban a kiszolgálás olyan „fázisokra" bomlik, amelyekben az igények csupán a csomóponttól, illetve a fázistól függő exponenciális eloszlást követő kiszolgálást kapnak — a fázisokat csomópontoknak tekintve — a már jól ismert hálózat áll előttünk. Kissé bonyolultabb az a helyzet, amikor a fázisokra jellemző paraméter magától az igénytől vagy a fázis állapotától függ. Ekkor ugyanis az egyes csomópontok nem egyformán viselkednek minden igényre nézve. Úgy is tekinthetjük, mintha több szintes kiszolgáló egységekből állna a hálózat, s a beérkező igény a rá érvényes paraméterek által meghatározott szintet választja. Tegyük fel, hogy egy igény adott, minden csomópontban előre meghatározott szintre lép. Ilyen modellt ír le KELLY a [19]-ben. Itt a kiszolgálás gamma eloszlást követ, ami azt jelenti, hogy a fázisokban azonos paraméterű exponenciális eloszlást kapnak az igények.

Az alapmodell ismertetésére a következő fejezetben térünk ki.

Újszerű a korábbiakhoz képest az [1]. Ebben a cikkben a szerzők megengedik, hogy minden igény csomópontról csomópontra megváltoztathassa a kiszolgálási

idejének eloszlását, melyről csak annyit kell feltenni, hogy racionális Laplace-transzformáltja létezzék. Ez épp azt jelenti, hogy minden fázis exponenciális eloszlású, nem feltétlen azonos paraméterrel. Így a rendszerben mozgó igények osztályokba sorolhatók aszerint, hogy éppen milyen eloszlású kiszolgálást kívánnak. Az osztályba sorolás — igénytípusok használata — KELLYnél is megtalálható az előbbihez hasonló, de avval nem teljesen azonos értelemben. Az [1] érdeme a fenti általánosításon túl, hogy az FCFS kiszolgálási elven kívül még háromféle prioritást is megenged. Mint majd látni fogjuk, Kelly modellje ebben a vonatkozásban jóval általánosabb.

A fenti modellekben a csomópontok szerepét átveszik a fázisok, így a faktorizáció is ez utóbbiak szerint történik, azaz a stacionárius eloszlásban a fázisok paramétereinek reciproka szorzódik, ezért nem jelenhetnek meg a csomópontokban történő kiszolgálások várható értékei. Az eddig tárgyalt modellek közösek abban, hogy állapotterük leírása „számosság jellegű", vagyis az egyes csomópontokban, illetve fázisokban tartózkodók száma az alapvető. Ez a megközelítés a leírás tekintetében egyszerűséget, de az eloszlások általánosításában komoly korlátokat jelent. Ennek feloldásáról a következő fejezetekben lesz szó.

5. Általános populációs modellek

Az előzőekben tárgyalt modellek igen sok lényeges megszorítást tartalmaztak akár a prioritási szabályokat, akár pedig a kiszolgálási idő eloszlását tekintjük. Az előbbi fejezetben már tettünk említést KELLY [20] modelljéről, ami mindkét tekintetben általánosabb feltételek mellett jut a szorzat formájú stacionárius eloszlásokhoz.

Tekintsünk most egy J csomópontú hálózatot, s tegyük fel, hogy a rendszerbe lépő igények I különböző módon járhatják be azt mielőtt távoznának. Ennek megfelelően fogjuk az igényeket I különböző típusba sorolni az általuk determinisztikusan választott útvonal szerint. Az i típusú igények $(i=1, \ldots, I)$ $\alpha(i)$ paraméterű *Poisson-folyamat* szerint érkeznek a rendszerbe és útvonaluk a következő:

$$r(i, 1),\ r(i, 2), \ldots, r(i, S(i)),$$

ahol $S(i)$ a felkeresett összes csomópontok száma, $r(i, s)$ pedig az a csomópont, amit egy i típusú igény útjának s-edik állomásaként felkeres. Tegyük fel, hogy az igények az egyes sorokban rendezetten állnak és a j-edik $(j=1, \ldots, J)$ sorban éppen n_j számú igény sorakozik.

A j-edik sor viselkedését a következő 4 pont írja le:

1) Az igények 1 paraméterű exponenciális eloszlású kiszolgálást kapnak.

2) Minden csomópontban egyetlen kiszolgáló van, aki $\Phi_j(n_j)$ intenzitással szolgálja ki az n_j igényt.

3) A j-edik sor l-edik pozícióján álló igény ennek a kapacitásnak a $\gamma_j(l, n_j)$ hányadát kapja. Ha a sorban az l-ediknek álló igény kiszolgálása befejeződik, úgy az $l+1, l+2, \ldots, n_j$ pozícióban tartozkodó igények egyet előre lépnek.

4) A j-edik sorba beérkező igény $\delta_j(l, n_j+1)$ valószínűséggel lép az l-edik pozícióra, s ekkor az $l, l+1, \ldots, n_j$-ik igény hátrább lép a sorban.

Természetesen

$$\sum_{i=1}^{n} \gamma_j(l, n) = 1,$$

$$\sum_{i=1}^{n} \delta_j(l, n) = 1.$$

Fel kell tenni, hogy nincsenek túlterhelt csomópontok, azaz $\Phi_j(n)>0$, ha $n>0$.

A $\gamma_j(\cdot, n_j)$ és $\delta_j(\cdot, n_j+1)$ függvények meghatározzák a csomópontban érvényes prioritási elvet. Ez azt jelenti, hogy ezek és a $\Phi_j(n_j)$ megfelelő megválasztásával lehet leírni. Példaként álljon itt néhány jól ismert kiszolgálási elv:

1) FCFS *(First Came First Served)*

$$\Phi_j(n_j) = 1$$

$$\gamma_j(l, n_j) = \begin{cases} 1, & l = 1 \\ 0, & \text{különben} \end{cases}$$

$$\delta_j(l, n_j+1) = \begin{cases} 1, & l = n_j+1 \\ 0, & \text{különben} \end{cases}$$

2) LCFSPR *(Last Came First Served Preemptive Resume)*

$$\Phi_j(n_j) = 1$$

$$\gamma_j(l, n_j) = \begin{cases} 1, & l = n_j \\ 0, & \text{különben} \end{cases}$$

$$\delta_j(l, n_j+1) = \begin{cases} 1, & l=n_j+1 \\ 0, & \text{különben} \end{cases}$$

3) PS *(Processor Sharing)*

$$\Phi_j(n_j) = 1$$

$$\gamma_j(l, n_j) = \frac{1}{n_j}, \quad l = 1, \ldots, n_j$$

$$\delta_j(l, n_j+1), \quad \text{tetszőleges} \quad l = 1, \ldots, n_j+1$$

4) IS *(Infinite Server)*

$$\Phi_j(n_j) = n_j$$

$$\gamma_j(l, n_j) = \frac{1}{n_j}, \quad l = 1, \ldots, n_j$$

$$\delta_j(l, n_j+1), \quad \text{tetszőleges} \quad l = 1, \ldots, n_j+1.$$

Visszatérve a modell leírására, jelölje $t_j(l)$ a j-edik sor l-edik pozíciójában álló igény típusát és $s_j(l)$ pedig a már érintett csomópontok számát (a jelenlegivel együtt).

A j-edik sor l-edik igénye a

$$c_j(l) = (t_j(l), s_j(l)) \tag{5.1}$$

rendezett párral jellemezhető egy adott időpontban. Így a teljes hálózat állapota egy változó sorhosszúságú mátrixszal adható meg, ahol a j-edik sor és l-edik oszlop kereszteződésében (5.1) áll.

$$\mathbf{C}(t) = (c_j(l))_{\substack{j=1,\ldots,J \\ l=1,\ldots,n_j}},$$

t az adott időt jelöli. A fenti mátrix egy *Markov-folyamatot* generál.

A $\mathbf{C}$ állapot megváltozása 3 módon lehetséges:

1) A k-adik sor m-edik pozíciójába lép kívülről egy i típusú igény. Az így megváltozott rendszer állapotát $\mathbf{T}^{im}\mathbf{C}$-vel jelöljük, k-t feltüntetni nem kell, hiszen azt i egyértelműen meghatározza. E változás intenzitása:

$$q(\mathbf{C}, \mathbf{T}^{im}\mathbf{C}) = \alpha(i)\delta_k(m, n_{k+1}).$$

2) A j-edik sor l-edik pozíciójában álló igény a k-adik sor m-edik pozíciójába lép át:

$$k = r(t_j(l), s_j(l)+1).$$

Az így megváltozott $\mathbf{C}$ állapotot $\mathbf{T}_{jlm}\mathbf{C}$-vel jelölve az átmenetintenzitás:

$$q(\mathbf{C}, \mathbf{T}_{jlm}\mathbf{C}) = \Phi_j(n_j)\gamma_j(l, n_j)\delta_k(m, n_k+1).$$

3) A j-edik sor l-edik pozíciójából kilépő igény a rendszert elhagyja, a hálózat megváltozott állapotát $\mathbf{T}_{jl}\mathbf{C}$ jelöli. Ebben az esetben az átmenetintenzitás:

$$q(\mathbf{C}, \mathbf{T}_{jl}\mathbf{C}) = \Phi_j(n_j)\gamma_j(l, n_j).$$

Látható, hogy — részben a típus használata miatt — ezek az intenzitások kevésbé általánosak, mint néhány korábban tárgyalt.

Most térjünk rá a stacionárius eloszlás vizsgálatára! E célból vezessük be a következő függvényeket:

$$\eta_j(i, s) = \begin{cases} \alpha(i), & r(i, s) = j \\ 0, & \text{egyébként} \end{cases}$$

$$a_j = \sum_{i=1}^{I} \sum_{s=1}^{S(i)} \eta_j(i, s)$$

a_j az egyensúlyban levő rendszer j-edik csomópontjába egységnyi idő alatt belépő igények száma. Tegyük fel, hogy

$$\sum_{n=0}^{\infty} \left(a_j^n \Big/ \prod_{l=1}^{n} \Phi_j(l) \right) < \infty, \quad j = 1, \ldots, J$$

5.1. TÉTEL: ([19a)] A fenti feltétel mellett a stacionárius eloszlás az alábbi:

$$\Pi(\mathbf{C}) = b \prod_{j=1}^{J} A_j(c_j),$$

ahol b normáló konstans és

$$A_j(c_j) = \prod_{l=1}^{n_j} (\eta_j(t_j(l), s_j(l)/\Phi_j(l)).$$

A prioritáson és a kiszolgálási időn kívül a routing is új típusú! Az eddigi — átmenetvalószínűség mátrixszal megadott — útvonalképzéssel ellentétben csak néhány lehetséges kombináció közül lehet választani, és ezek meghatározott hosszúságúak. (A klasszikus modelleknél tetszőleges hosszú utak is előfordulhatnak pozitív valószínűséggel.) Egy korábbi cikkben [19] egyébként ezek az eredmények a klasszikus útképzésre (routing) is be vannak bizonyítva.

Érdemes megvizsgálni a további általánosítási lehetőségeket. COHEN [8] olyan zárt modellt vizsgált, ahol M igény vándorol P csomópont között az i ($i=1, \ldots, M$) igény a j-edik ($j=1, \ldots, P$) csomópontban $u_i^{(j)}$ időt igényel. $u_i^{(j)}$ valószínűségi változó. $x_t(j)$ jelöli a t időpontban a j-edik kiszolgálóegységben tartózkodó igények számát.

$$x_t(1)+x_t(2)+\ldots+x_t(P) = M$$

$$x_t := (x_t(1), \ldots, x_t(P)),$$

$$x_t(j) \geqq 0, \quad j = 1, \ldots, P.$$

A kiszolgálási elv minden csomópontban általánosított *processor sharing*, azaz ha a j-edik csomópontban x igény van a t időpontban, akkor ott minden igény a $(t, t+\Delta t)$ idő alatt $f^{(j)}(x)\Delta t$ kiszolgálás mennyiséget kap.

$$0 \leqq f^{(j)}(x) < \infty$$

$$xf^{(j)}(x) \leqq K(j), \quad x = 1, \ldots, M$$

$$\Phi^{(j)}(x) := \begin{cases} \left\{ \prod_{k=1}^{x} f^{(j)}(k) \right\}^{-1}, & x = 1, \ldots, M \\ 0, & x = 0 \end{cases}$$

$K(j)$ a j-edik csomópont teljes kapacitása. Az irányításmátrix az i igényre $[p_i^{(h,j)}]_{h,j=1,\ldots,P}$.

Egy igény által a különböző kiszolgálási egységekben igényelt időmennyiségek legyenek független valószínűségi változók abszolút folytonos $B_i^{(j)}(\cdot)$ eloszlásfüggvénnyel:

$$B_i^{(j)}(\tau) = P(u_i^{(j)} < \tau), \quad i = 1, \ldots, M; \; j = 1, \ldots, P; \; \tau > 0,$$

$$B_i^{(j)}(0+) = 0.$$

Korábban az exponenciális eloszlás „örökifjú" tulajdonsága miatt az állapotváltozások valószínűségei minden t időpontban azonosak voltak. A jelen modellben erre nem számíthatunk. Az állapotváltozások valószínűségei az igények kiszolgáltsági fokától függenek. Ezért nem elegendő ismerni az M igény elhelyezkedését a P csomópontban, hanem tudni kell az általuk igényelt kiszolgálás még hátralevő részének nagyságát is. Ez utóbbi — minthogy *processor sharing* van — folyamatosan csökken az idő múlásával, míg csak el nem éri a nullát.

$\lambda(j, k)$-val jelölve azt az igényt, amelyik a t időpontban a j-edik sor k-adik pozíciójában áll, a $\lambda(x)$ változó sorhosszúságú mátrixszal lehet megadni, hogy az egyes sorokban „hol" „ki" tartózkodik.

$$\lambda(x) = [\lambda(j, k)] \quad k = 1, \ldots, x(j), \; j = 1, \ldots, P$$

t időpontban a $\lambda(j, k)$ igény hátralevő kiszolgálási idejét $\tau(j, k)$, s a már megszerzettet $\sigma(j, k)$ jelöli.

$$\tau(\lambda(x)) = [\tau(j, k)] \quad k = 1, \ldots, x(j), \; j = 1, \ldots, P$$

$$\sigma(\lambda(x)) = [\sigma(j, k)] \quad k = 1, \ldots, x(j), \; j = 1, \ldots, P$$

$X_t = \{x_t, \lambda(x_t), \tau(\lambda(x_t))\}$ sztochasztikus folyamattal írható tehát le a modell.

X_t markovitásához elengedhetetlen a *processor sharing* feltételezése, azaz a τ folytonos változása.

Látható, hogy ha τ-t elhagynánk, s a már megszokott módon — most $(x_t, \lambda(x_t))$-vel jelölve — akarnánk megadni a hálózatot leíró sztochasztikus folyamatot, csak egy *fél-Markov folyamathoz* jutnánk. Δt idő alatt az állapotváltozások kétfélék lehetnek:

a) valamelyik csomópontban egy igény kiszolgálása befejeződik, ami azt jelenti, hogy ez az igény egy másik csomópontot keres fel, vagy
b) egyik τ sem éri el a 0-t ez alatt az idő alatt.

Ezek a változások integro-differenciál egyenletekkel írhatók le, melyeket a stacionárius eloszlásoknak nyilván ki kell elégíteniük. A $p(t, x, \lambda(x), \tau(\lambda(x)))$ stacionárius eloszlásfüggvények olyanok, hogy a $B_i^{(j)}(\cdot)$-k más paramétere, mint a várható érték, nem szerepel bennük.

Ez éppen azt jelenti, hogy a $p(\cdot, \cdot, \cdot, \cdot)$-k nem függnek attól, hogy az egyes igények milyen eloszlású kiszolgálást igényelnek. Azon az áron, hogy egy új, általánosabb típusú állapotteret definiáltunk egy szigorúan csökkenő folytonos (kiegészítő) változó is bevezetésre került. Ezek az állapotok már egyáltalán nem emlékeztetnek a klasszikus modellek leírására.

Más szerzők is hasonló eredményre jutottak ([5]), mint COHEN. E cikk felépítése, módszerei azonban teljesen más jellegűek. A sorok viselkedését a hálózat más részeitől függetlenül vizsgálja, és azokat háromféle egyensúlyi egyenlettel jellemzi. Az ily módon kapott eredményekből következtet az egész rendszer működésére. A szerzők megmutatják, hogy ha a kiszolgálás nem exponenciális eloszlású, akkor csak azon kiszolgálási elvek mellett lehet a stacionárius eloszlás „szorzat formájú", ahol az igények kiszolgálása azok beérkeztével azonnal megkezdődik, vagy nincs várakozás (a *processor sharing* éppen ilyen). [28]-ból is kiolvasható, hogy a *processor sharing* sokkal kényelmesebben kezelhető az igazi várakozásos struktúráknál.

Láttuk tehát, hogy a kiszolgálás eloszlására tett exponencialitási feltétel jelentősen általánosítható. Ezek az általánosítások egy teljesen új típusú állapottér következményei. Ezekben az esetekben a soroknak nemcsak a népessége írta le a hálózatot, hanem annak is szerepe volt, hogy a sorokban milyen igények és hogyan helyezkednek el. Ez természetesen egy sokkal bonyolultabb állapotteret eredményezett, de végső soron ugyanolyan egyszerűen számítható algoritmusokat kaptunk, mint a klasszikus esetekben.

6. Más megközelítések

Az eddigi eredményeket röviden összefoglalva megállapíthatjuk, hogy igyekeztünk olyan állapottereket konstruálni, hogy a rajtuk definiált sztochasztikus folyamatok *Markov-láncok* legyenek. Ezek után olyan további megszorításokkal éltünk, hogy a *Kolmogorov-egyenleteknél* részletesebb egyensúlyi feltételek teljesüljenek, és az esetleg igen nagy ismeretlenszámú egyenletrendszerek könnyen megoldhatók legyenek. Minden esetben lényegében explicit megoldásokat írtunk fel szorzatformában.

Egyéb lehetőségek is vezethetnek célra. Néhány kevésbé ismert próbálkozást is összefoglalunk. Ezek mindegyikéről önálló összefoglalót lehetne írni, ami viszont nem érné meg biztosan a fáradságot. A teljesség kedvéért soroljuk fel a kevésbé ismert próbálkozásokat, elsősorban továbbgondolásra ösztönzendő. Természetesen ezek nem kiforrott szintézisek, az eddigiekhez hasonló egységes szemléletmód és matematikai megalapozottság többnyire még várat magára.

6.1 Állapottér összevonás (lumping, collapsing) [2], [29]

S állapottér számosságát csökkenthetjük azáltal, hogy definiáljuk $S=\bigcup_{i=1}^{K} A_i$ partíciót. Ily módon az S-en definiált $\xi(t)$ *Markov-lánc* aszimptotikus viselkedése meghatároz egy $\eta(t)$ sztochasztikus folyamatot $P\{\eta(t)=j\}=\sum_{i\in A_j} P\{\xi(t)=i\}$ definícióval. $\eta(t)$ nem lesz feltétlenül *Markov-lánc* ([2]), de

$$y_j = \sum_{i=S_j} \lim_{t\to\infty} P\{\xi(t)=i\} \tag{6.1}$$

definiálja a stacionárius eloszlását. Nagyon sokszor elég lenne $\mathbf{y}$ ismerete is $\xi(t)$ stacionárius eloszlása (π) helyett.

6.1. TÉTEL: Létezik $\mathbf{Q}$ átmenet valószínűség mátrix, hogy $\mathbf{y}=\mathbf{y},\mathbf{Q}$, ahol $\mathbf{y}$-t (6.1) adja meg. $\mathbf{Q}$ elemei $\mathbf{P}$ és π segítségével kifejezhetők:

$$q_{ij} = \sum_{l\in A_i} \pi_i \Big(\sum_{s\in A_i} \pi_s\Big)^{-1} \sum_{r\in A_j} p_{lr}. \tag{6.2}$$

Bizonyítás. Egyszerűség kedvéért csak $S=\{0,1,\ldots,K\}$, $A_1=\{0,1\}$, $A_i=\{i\}$ esetet látjuk be. Definiáljuk a $\mathbf{B}$ mátrixot:

$$b_{i1} = p_{i0}+p_{i1}, \quad \text{ha} \quad i \geqq 2$$
$$b_{ij} = p_{ij} \quad , \quad \text{ha} \quad i \geqq 2,\ j \geqq 2$$

és b_{1j}-t az

$$\mathbf{y} = \mathbf{y}\cdot\mathbf{B} \tag{6.3}$$

egyenletrendszerből határozzuk meg. (Ez K darab egyenlet a K ismeretlenre: b_{1j}, $j=1,\ldots,K$). $\sum_{j=1}^{K} b_{1j}=1$-gyel ez egy korrekt feladat, ha létezik π stacionárius eloszlás.

$\pi=\pi\mathbf{P}$ első két egyenletét összeadva, és a többit változatlanul hagyva kapjuk

$$\begin{aligned} &\pi_0(p_{00}+p_{01})\pi(p_{10}+p_{11})+\ldots+\pi_n(p_{n0}+p_{n1}) = \pi_0+\pi_1 \\ &\pi_0 p_{02}+\pi_1 p_{12}+\ldots+\pi_n p_{n2} \qquad\qquad = \pi_2 \\ &\qquad\vdots \qquad\qquad\qquad\qquad\qquad\qquad\quad \vdots \\ &\pi_0 p_{0n}+\pi_1 p_{1n}+\ldots+\pi_n p_{nn} \qquad\qquad = \pi_n \end{aligned} \tag{6.4}$$

Kivonva (6.3)-at (6.4)-ből

$$\begin{aligned}
&\pi_0(p_{00}+p_{01})+\pi_1(p_{10}+p_{11}) = (\pi_0+\pi_1)b_{11}\\
&\pi_0 p_{02} \qquad +\pi_1 p_{12} \qquad = (\pi_0+\pi_1)b_{12}\\
&\quad\vdots \qquad\qquad\quad \vdots\\
&\pi_0 p_{0n} \qquad +\pi_1 p_{1n} \qquad = (\pi_0+\pi_1)b_{1n}
\end{aligned}$$

adódik. Ha ebből b_{1i}-ket kifejezzük, **Q** és **B** egybeesése már látszik. Tetszőleges partícióra a bizonyítás teljesen hasonló.

Megjegyzések. 1. (6.2)-ben q_{ij} **P** soraiból vett részletösszegek konvex lineáris kombinációja, ahol a súlyok a soroknak megfelelő stacionárius valószínűségeivel arányosak. Ismeretesek eljárások, melyekben a stacionárius eloszlást azon feltétel mellett számoljuk, hogy a lánc A_i-ben van az állapottéren belül: $\lim_{t\to\infty} P\{\xi(t)=j|\xi(t)\in S_i\}$. Ily módon a súlyok megadhatók, de az eljárás — általában — nem kedvezőbb a hagyományos *Gauss-eliminációnál.*

2. Fenti tétel általánosítása a [2]-beli klasszikus eredménynek. Nevezetesen ha $\sum_{r\in S_i} p_{lr}=C_j^{(i)}$ minden $l\in S_j$-re, akkor $q_{ij}=C_j^{(i)}$. Ilyenkor a lineáris kombinációban az elemek (a részletösszegek) egyenlők, tehát a súlyok tetszőlegesek lehetnek.

3. A tétel jelentősége elsősorban abban van, hogy lehetőséget nyújt **Q** elemeinek közelítésére.

Ez a következőképpen történik:

p eloszlást nagyobbnak mondjuk mint **q** (**p**$\succ$**q**), ha minden k-ra

$$\sum_{i=1}^{k} p_i \leqq \sum_{i=1}^{k} q_i.$$

Ugyanígy **A** és **B** sztochasztikus mátrixok között **A**$\succ$**B** áll fenn, ha minden sorukra (mint eloszlásra) érvényes $\mathbf{a}_i\succ\mathbf{b}_i$. Bizonyos feltételek mellett — amelyek populációs modellekre általában teljesülnek — könnyű **A**, **B** sztochasztikus mátrixokat konstruálni, hogy **A**$\succ$**Q**$\succ$**B** teljesüljön, amiből **u**=**uA** és **v**=**vB** fixpontjaikra **u**$\succ$**y**$\succ$**v** következik. Ezt az eljárást egy példával illusztrálja [29].

6.2 *Dekompozíciós eljárások (aggregation, decomposition)* ([9]), [23], [4])

Ezen belül két irányzatot kell megkülönböztetni:

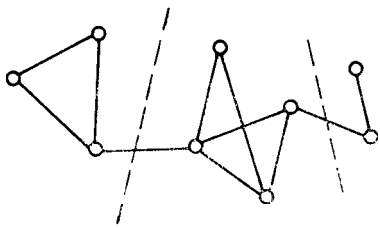
i) Ismét az állapotteret bontjuk részhalmazokra, de most a modell meghatározta koncepció szerint: $S=\bigcup_{i=1}^{L} S_i$. Az egyes részhalmazok közötti átmenetek elhanyagolhatók. Ez azt jelenti, hogy az eredeti átmenet valószínűség mátrixot

$$\mathbf{P} = \mathbf{P}_0+\mathbf{P}_1$$

alakban állítjuk elő, ahol $\mathbf{P}_0$ a fődiagonális menti blokkokból áll, $\mathbf{P}_1$ pedig ritka mátrix, és elemei kicsik. $\mathbf{P}_0$ *reducibilis Markov-láncot* definiál, de ennek aszimptotikus viselkedése jóval egyszerűbben határozható meg. Ha $\mathbf{P}_1$-re további feltételek teljesül-

nek, akkor ez az eljárás jól közelíti az eredeti folyamat viselkedését ([9]). [26] egy példával illusztrálja ezt a módszert.

ii) Több szerző ([3], [4]) javasolja a tárgyalás következő egyszerűsítését:



A hálózatot egyszerűen feldaraboljuk, és az egyes részeket önmagukban vizsgáljuk, majd az egészet úgy, hogy a részeket egyedi csomópontoknak tekintjük. (Fenti elrendezésből egy három csomópontú rendszer lesz.) Itt azonban a részrendszerek paramétereit csak az eredeti megoldásból tudjuk kiszámítani (l. 6.1). Közelítő megoldásnál esetleg segíthet ez az ötlet: a 6.1-beli felbontás (partíció) alapját képezheti például az, hogy az egyes részekben levő populációkat összevonjuk.

A sokat emlegetett *„Norton-tétel analógiájának"* valamint a *„flow-eigenvalue"*-nak csak elméleti jelentősége van: Valóban egy tetszőleges részhálózathoz létezik olyan csomópont, amely vele ekvivalens, de ennek paramétereit (bemenő intenzitás, kiszolgálási intenzitás stb.) általában csak a teljes megoldás ismeretében tudjuk kiszámítani.

6.3 Működés szerinti vizsgálat (operational analysis) ([10]).

Már említettük, hogy a feltételek — amelyek a matematikai tárgyalást könnyítették — nem biztos, hogy teljesülnek a valóságban, és ellenőrzésük mindenképpen nehézkes.

[10]-beli szerzők egy új típusú vizsgálatot javasolnak, ahol a kiindulási paraméterek jobban alkalmazkodnak a mérési és megfigyelési lehetőségekhez. Az említett cikk tulajdonképpen ideológiai alapot nyújt a klasszikus eredményekhez (különféle egyensúlyok). Megjelenik néhány új paraméter (kihasználtság, feltételes események, várakozási idők stb.) és kétségkívül egy átfogóbb leírás körvonalazódik. Az alkalmazott algoritmusok és összefüggések azonban nem sok újat jelentenek.

Teljesen jogos a szerzők észrevétele, miszerint a hálózatok tárgyalását inkább az elmélet (*queueing theory*) hagyományai, mintsem a valódi igények és lehetőségek befolyásolták. Méréstechnikai szempontból ugyanis sokkal nehezebb egy időeloszlást megfigyelni és illeszteni, mint diszkrét elemeket vagy eseményeket számlálni. Ez azt jelenti, hogy sokkal információdúsabb lehet mondjuk a sorhosszból levezetni a kiszolgálási időket, vagy szerencsésebb lenne pontfolyamatokat úgy jellemezni, hogy megmondjuk annak a $P(T, k)$ valószínűségét, hogy rögzített T idő alatt éppen k esemény következik be. Minthogy erre alig találunk példát, valóban egy új típusú tárgyalás lenne. A valódi nehézséget — főleg hazánkban — például abban keresnénk, hogy kevés az olyan működő — tehát mérhető és megfigyelhető — számítógép-hálózat, amelynél a legfőbb gondot az *optimális* működés jelentené.

6.4 Átlagokon alapuló vizsgálat (mean value analysis) ([25], [25a])

Kétségtelen, hogy a szorzatformában adott képletekben ((4.8), (4.10)) a kiszolgálási időnek csak a várható értéke szerepel. Ez exponenciális eloszlások esetében nem meglepő, hiszen egyparaméteres lévén ez megszokott a kiszolgálás elméletében. Még [8] esetben is indokolja ezt az a tény, hogy — mivel nincs kiszolgálás — érvényesül a fél-markovi jelleg. De várakozásos rendszereknél általános esetben még a legkedvezőbb feltételek mellett sem okvetlenül ez a helyzet. Ismert ugyanis ([22]), hogy egyetlen csomópont esetén tetszőleges kiszolgálási eloszlás mellett — még *Poisson-bemenet* esetén is — a sorhossz várható értéke a kiszolgálási idő második momentumától is függ:

$$E(Q) = \varrho + \varrho^2 \frac{1+C^2}{2(1-\varrho)},$$

ahol ϱ a kiszolgálás várható ideje szorozva a bemenő intenzitással, C pedig a kiszolgálási idő relatív szórásnégyzete.

Létezik az igen általános érvényű *Little-formula*

$$\lambda \cdot T = N$$

ami azt mondja ki, hogy a bemenő intenzitás és a rendszerben töltött várható időnek a szorzata megegyezik a rendszerben levők számának várható értékével. Ez lehetőséget ad, hogy a populációszám várható értékéből a késleltetési idő átlagára következtessünk.

[25] sokat ígérő címe ellenére egyáltalában nem új hozzáállás, és inkább zárt modellek számítási algoritmusaként jelentős, mintsem új eleméletként. A vizsgálat kiterjed a késleltetési idő várható értékére és az átbocsátóképességre is, de a hagyományosnál nem lényegesen egyszerűbben. A momentumok az egyes valószínűségek részletes kiszámítása nélkül adódnak [25a]-ban, tehát a sok gondot okozó normáló konstansokra nincs szükség. Az algoritmus azonban rekurzív, vagyis egy (M, K) hálózathoz (M csomópont, és pontosan K igény) valamennyi (M, n), $n=0, 1, \ldots, K-1$ modell kiértékelésére szükség van. A tárgyalt eset elég speciális, minthogy $\varphi_i(n_i) = \mu_i$, illetve $\varphi_i(n_i) = \min(n_i, k_i)$.

6.5 Diffúziós közelítések ([13], [14])

Elsősorban egyedi kiszolgálórendszerek nagy terhelését lehet jól közelíteni tetszőleges eloszlású kiszolgálási időeloszlás mellett. Két csomópontot tartalmazó hálózatokra vonatkozó általánosítását is elvégezték ([14]). Nagyobb hálózatok esetében ez sokkal bonyolultabb. Ilyenkor az elmélet igen érdekes lehet, de a pontosság is esetleges. Külön nehézséget okoz, hogy először diszkrét változókhoz kell eloszlásfüggvényeket illeszteni, majd a folytonos eredményeket ismét diszkrét állapottéren interpretálni.

6.6 Hálóelmélet és optimalizálás ([11], [26])

Az operációkutatás eszközeinek a konkrét tervezéshez közvetlenül történő felhasználásáról kiváló összefoglaló [27]. Erre az ugyancsak szépen kidolgozott területre nem volt célunk kalandozni. Megemlítjük viszont azokat a kísérleteket, amelyek a populációs modellek és az optimalizálás összehasonlítását célozzák. A legegyszerűbb exponenciális esetre (4.6 tételben $a(|\mathbf{n}|)=1$, $\varphi_i(n_i)=1$, $f(\mathbf{n})=1$) a késési idő explicite kifejezhető. Ezt próbálják minimalizálni az áramlási paraméterek (p_{ij}) függvényében [26] és [11].

Végül hálánkat fejezzük ki mindazoknak, akik észrevételeikkel segítettek. Elsősorban a lektoroknak, valamint ARATÓ MÁTYÁS professzornak és KERTÉSZ ZSUZSÁNAK tartozunk köszönettel.

IRODALOM

[1] BASKETT, F., CHANDY, K. M., MUNTZ, R. R. and PALACIOS, F. G., "Open, closed, and mixed networks of queues with different classes of customers", *J. ACM* **22** (1975) 248—260.
[2] BURKE, C. J. and M. ROSENBLATT, "A Markovian function of a Markov chain", *Ann. Math. Stat.* **29** (1958) 1112—1122.
[3] CHANDY, K. M., HERZOG, U. and WOO, L., "Parametric analysis of general queueing networks", *IBM J. Res. Dev.* **19** (1975) 36—42.
[4] CHANDY, K. M., and C. H. SAUER, "Approximate methods for analyzing queueing networks-models", *Comp. Surveys* **10** (1978) 281—317.
[5] CHANDY, K. M., HOWARD, J. H. and TOWSLEY, D. F., "Product form and local balance in queueing networks", *J. ACM* **24** (1977) 250—263.
[6] CHUNG, K. L., *Markov Chains with Statoniary Transition Probabilities* (Springer, Berlin, 1960).
[7] COHEK, J. E., "Markov population processes as models of primate social and population dinamics", *Theor. Popul. Biol.* **9** (1972) 119—134.
[8] COHEN, J. W., "Multiple phase service network", *Acta Inform.* **12** (1979) 245—284.
[9] CURTOIS, P-J., "Error analysis in nearly completely decomposable stochastic systems", *Econometrica* **43** (1975) 691—709.
[10] DENNING, P. J. and J. P. BUZEN, "Operational analysis of queueing networks", *Comp. Surveys* **10** (1978) 225—261.
[10a] DISNEY, R. L., "Random flow in a queueing network, a review and critique", *AIIE Trans.* **7** (1975) 268—288.
[11] FRATT, L., GERLA, M. and KLEINROCK, L., "The flow deviation method", *Networks* **3** (1979) 97—133.
[12] GANS, P. J. "Open first-order stochastic processes", *J. Chem. Phys.* **33** (1960) 691—694.
[13] GAVER, D. F. and G. S. SHEDLER, "Processor utilization in multiprogramming systems via diffusion approximation", *Opns. Res.* **21** (1973) 569—576.
[14] GELENBE, E., "On approximate computer system models", *Comp. Archit. and Networks Eds. E. Gelenbe / R. Mahl*, North-Holl. 1974.
[15] GORDON, W. I. and G. F. NEWELL, "Closed queueing systems with exponential servers", *Opns. Res.* **13** (1965) 254—265.
[16] JACKSON, J. R., "Networks of waiting lines", *Opns, Res.* **5** (1957) 518—421.
[17] JACKSON, J. R., "Jobshop-like queueing systems", *Management Science* **10** (1963) 131—142.
[18] KARLIK, S., *First Course in Stochastic Processes* (Academic Press, New York and London, 1968).
[19] KELLY, F. P., "Networks of queues with customers of different types", *J. Appl. Prob.* **12** (1975) 542—554.
[20] KELLY, F. P., "Networks of queues", *J. Appl. Prob.* **8** (1976) 416—432.
[21] KINGMAN, J. F. B., "Markov population processes", *J. Appl. Prob.* **6** (1969) 1—16.
[22] KLEINROCK, L., *Sorbanállás, kiszolgálás* (Műszaki Kiadó, Budapest, 1979).
[23] KUN, I., *„Zur Bedienungstheoretischen Modellierung"* MO/13, Sztaki Working Paper, 1980.
[24] REICH, E., "Waiting times when queues are in tandem", *Ann. Math. Stat.* **28** (1957) 768—773.

[25] REISER, M., "Mean value analysis of queueing networks, a new look at an old problem", *4th Symp. on Modelling and Pref. Eval. of Comp. Systems,* Vienna, 1979.
[25a] REISER, M. and LAVENBERG, S. S., "Mean value analysis of closed multichain queueing networks", *J. of ACM* **27** (1980) 313—322.
[26] SUGÁR, P., Számítógép-hálózatok erőforrásainak hozzárendelései, Kandidátusi disszertáció, Budapest, 1980.
[27] SZ. TURCHÁNYI, P., „Csomagkapcsolt számítógép-hálózatok tervezése", *Alk. Mat. Lapok* **4** (1978) 245—270.
[28] TOMKÓ, J., „Számítógépek központi egységének kihasználtságáról, II." *Alk. Mat. Lapok* **3** (1977) 83—96.
[29] TÖRÖK, L. T. and MESSING, GY., "System-bus load investigations", KFKI-Report, Budapest, 1980.
[30] WHITTLE, P., "Nonlinear migration processes", *Proc. 36th Session of the Int. Stat. Inst.*, 1967.
[31] WHITTLE, P., "Equilibrium distribution for an open migration process", *J. Appl. Prob.* **5** (1968) 567—571.

(Beérkezett: 1980. november 27.)

GÁRDOS ÉVA
KÖZPONTI STATISZTIKAI HIVATAL
1525 BUDAPEST, PF. 51.

TÖRÖK TURUL
MTA KÖZPONTI FIZIKAI KUTATÓ INTÉZET
1525 BUDAPEST, PF. 49.

POPULATION PROCESSES AND COMPUTER NETWORKS

É. GÁRDOS and T. TÖRÖK

Population processes seem to be effective tools in several fields of applied mathematics. If individuals or discrete quantities migrate among places mutually connected with each other, these processes can describe the complex structure.

An attempt was made to sum up results applicable for computer network models. Not only classical results are listed, but further possibilities and ideas are sketched as well.

Alkalmazott Matematikai Lapok 6 (1980) 313—322

RELÁCIÓS ADATMODELL FUNKCIONÁLIS FÜGGŐSÉGEINEK ÁLTALÁNOSÍTÁSA

DEMETROVICS JÁNOS ÉS GYEPESI GYÖRGY

Budapest

A dolgozatban relációs adatmodell funkcionális függőségeivel és azok három általánosításával foglalkozunk: a duális, az erős és a gyenge függőségekkel. Ezen függőségek mindegyikének a teljes családját axiomatizáljuk (funkcionális függőségek teljes családjait először az [1] dolgozatban axiomatizálták, majd ugyanezt a duális és az erős függőségekre a [8] dolgozatban adták meg; a gyenge függőségek teljes családjainak az axiomatizálása új eredmény).

Axiómáink a mátrixok egyenlőséghalmazainak kombinatorikus jellemzésén alapulnak. Bebizonyítunk egy lényeges különbséget is a gyenge és a többi függőség között.

1. Bevezetés

Az E. F. CODD [6] által bevezetett relációs adatmodell az adatkezelés egyik legígéretesebb eszköze. Ez a modell az adattárolást szemléletes formában, mátrix alakban valósítja meg. A mátrix sorai az adatrekordok, az oszlopok pedig a tulajdonságok, más szóval az attributumok.

A pontos definíció a következő. Legyen Ω nem üres véges halmaz ($\Omega=\{a_1, \ldots, a_n\}$). Ω feletti relációnak (R) nevezzük az Ω-n értelmezett függvények véges halmazait. Tehát, ha R egy reláció Ω felett, és $f\in R$, akkor f az Ω-n értelmezett függvény. A relációkat szemléletesen kétdimenziós táblázatként ábrázoljuk: ha R reláció Ω felett és $R=\{f_1, \ldots, f_k\}$, akkor R táblázata:

	a_1	a_2	...	a_n
f_1	$f_1(a_1)$	$f_1(a_2)$	...	$f_1(a_n)$
⋮	⋮	⋮		⋮
f_k	$f_k(a_1)$	$f_k(a_2)$	...	$f_k(a_n)$

R elemeit R sorainak nevezzük.

Egy relációs adatbázis feladata általánosan fogalmazva az információszolgáltatás. Ennek hatékony eszköze az adatok közötti összefüggések feltárása. Relációs adatmodell alkalmazásakor ezen összefüggések egyik fontos formája az E. F. CODD [7] által bevezetett *funkcionális függés.*

DEFINÍCIÓ: Legyen Ω véges, nem üres halmaz és R egy reláció Ω felett. Legyenek továbbá A és B részhalmazai Ω-nak. Azt mondjuk, hogy *B funkcionálisan függ A-tól*

R-ben (és ezt $A \xrightarrow[R]{f} B$-vel jelöljük), ha R bármely két h, g sorára igaz:

$$(1.1) \qquad (\forall a \in A)(h(a) = g(a)) \to (\forall b \in B)(h(b) = g(b)).$$

Az (1.1) formulában szereplő kvantorok lehetséges változtatásait végrehajtva nyerhetünk további függés fogalmakat. Az előző definíció jelöléseit használva definiáljuk a *duális, erős* és *gyenge* függéseket:

B duálisan függ A-tól R-ben (jelölése $A \xrightarrow[R]{d} B$), ha R bármely két h, g sorára

$$(1.2) \qquad (\exists a \in A)(h(a) = g(a)) \to (\exists b \in B)(g(b) = h(b)).$$

B erősen függ A-tól R-ben (jelölése $A \xrightarrow[R]{s} B$), ha R bármely két h, g sorára

$$(1.3) \qquad (\exists a \in A)(h(a) = g(a)) \to (\forall b \in B)(h(b) = g(b)).$$

B gyengén függ A-tól R-ben (jelölése $A \xrightarrow[R]{w} B$), ha R bármely két h, g sorára

$$(1.4) \qquad (\forall a \in A)(h(a) = g(a)) \to (\exists b \in B)(h(b) = g(b)).$$

Legyen R egy reláció Ω felett. Vezessük be az R-ben fennálló funkcionális, duális, erős, ill. gyenge függések halmazára a következő jelöléseket:

$$\mathscr{F}_R = \{(A, B)\colon A \subseteq \Omega,\ B \subseteq \Omega \ \&\ A \xrightarrow[R]{f} B\};$$

$$\mathscr{D}_R = \{(A, B)\colon A \subseteq \Omega,\ B \subseteq \Omega \ \&\ A \xrightarrow[R]{d} B\};$$

$$\mathscr{S}_R = \{(A, B)\colon A \subseteq \Omega,\ B \subseteq \Omega \ \&\ A \xrightarrow[R]{s} B\};$$

$$\mathscr{W}_R = \{(A, B)\colon A \subseteq \Omega,\ B \subseteq \Omega \ \&\ A \xrightarrow[R]{w} B\}.$$

Legyen $\mathscr{Y} \subseteq P(\Omega) \times P(\Omega)$. Azt mondjuk, hogy $\mathscr{Y}$ teljes f-család (d-család, s-család, ill. w-család), ha létezik olyan R reláció Ω felett, amelyre $\mathscr{Y} = \mathscr{F}_R$ ($\mathscr{Y} = \mathscr{D}_R$, $\mathscr{Y} = \mathscr{S}_R$, $\mathscr{Y} = \mathscr{W}_R$ rendre).

Természetes igény leírni azokat a tulajdonságokat, melyek a reprezentáló reláció konkrét választásától függetlenül fennállnak az egyes teljes családok struktúrájában. Ebben az irányban az első fontos lépést W. W. ARMSTRONG tette [1], aki jellemezte a teljes f-családokat. Axiómarendszere logikai természetű abban az értelemben, hogy a funkcionális függés definíciójára támaszkodik és nem a teljes f-családok kombinatorikus struktúráját vizsgálja. Bevezetett néhány fontos, a funkcionális függések gyakorlati alkalmazása szempontjából is hasznos fogalmat: a maximális függés, a maximális jobb oldal és a teljes család generálásának fogalmát. Fontos (és első pillantásra meglepő) eredménye, hogy tetszőleges teljes f-családot a maximális függőségeinek jobb oldalai meghatározzák.

A teljes d-családokat és a teljes s-családokat vizsgálja a [8] dolgozat és ARMSTRONGéval analóg axiómarendszereket közöl ezek jellemzésére.

A funkcionális függések elméletében hamar jelentkeztek azok a problémák, melyek a teljes f-családok kombinatorikus struktúrájára vonatkoznak [2], [5], [9]. [10]-ben rámutatunk arra, hogy még az annyira logikai természetűnek látszó fogalom is kombinatorikai hátterű, mint a generálás.

Ebben a cikkben új axiómákat adunk a teljes *f*-, *d*- és *s*-családok jellemzésére és axiomatizáljuk az eddig még nem jellemzett teljes *w*-családokat.

Úgy gondoljuk, hogy axiómáink rámutatnak a teljes családok struktúrájának kombinatorikus jellemzőire ([10]-ben ezt használva bizonyítunk ilyen természetű tételeket); továbbá mutatunk egy lényeges különbséget a gyenge függés és a többi között. Ez a különbség lényegében az, hogy az $\emptyset$ kezdőtagú függések csak a gyenge függésnél mutatnak más tulajdonságokat, mint a nem üres kezdőtagúak.

2. A teljes *f*-, *d*- és *s*-családok leírása

Ebben a részben új axiómákat adunk a teljes *f*-, *d*- és *s*-családok jellemzésére, és megfogalmazunk egy axiómát olyan teljes *w*-családokra, melyek nem tartalmaznak üres kezdőtagú függést. Bebizonyítjuk, hogy axiómáink ekvivalensek a nekik megfelelő „régiekkel". Az axiómák hasonlóságának igazi okát a 3. pontban írjuk le és ott axiomatizáljuk a teljes *w*-családokat.

A teljesség kedvéért leírjuk a teljes *f*-, *d*- és *s*-családok [1]-ben és [8]-ban megadott axiómarendszereit:

f-axiómák:

(F1) $(\forall A \subseteq \Omega)((A, A)\in\mathscr{F})$;

(F2) $(\forall A, B, C \subseteq \Omega)((A, B)\in\mathscr{F} \ \& \ (B, C)\in\mathscr{F} \to (A, C)\in\mathscr{F})$;

(F3) $(\forall A, B, C, D \subseteq \Omega)((A, B)\in\mathscr{F} \ \& \ C \supseteq A \ \& \ D \subseteq B \to (C, D)\in\mathscr{F})$;

(F4) $(\forall A, B, C, D \subseteq \Omega)((A, B)\in\mathscr{F} \ \& \ (C, D)\in\mathscr{F} \to (A\cup C, B\cup D)\in\mathscr{F})$.

ϑ-axiómák:

(D1) $(\forall A \subseteq \Omega)((A, A)\in\mathscr{D})$;

(D2) $(\forall A, B, C \subseteq \Omega)((A, B)\in\mathscr{D} \ \& \ (B, C)\in\mathscr{D} \to (A, C)\in\mathscr{D})$;

(D3) $(\forall A, B, C, D \subseteq \Omega)((A, B)\in\mathscr{D} \ \& \ C \subseteq A \ \& \ B \subseteq D \to (C, D)\in\mathscr{D})$;

(D4) $(\forall A, B, C, D \subseteq \Omega)((A, B)\in\mathscr{D} \ \& \ (C, D)\in\mathscr{D} \to (A\cup C, B\cup D)\in\mathscr{D})$;

(D5) $(\forall A \subseteq \Omega)((A, \emptyset)\in\mathscr{D} \to A = \emptyset)$.

γ-axiómák:

(S1) $(\forall a\in\Omega)((\{a\}, \{a\})\in\mathscr{S})$;

(S2) $(\forall A, B, C \subseteq \Omega)((A, B)\in\mathscr{S} \ \& \ (B, C)\in\mathscr{S} \ \& \ B \neq \emptyset \to (A, C)\in\mathscr{S})$;

(S3) $(\forall A, B, C, D \subseteq \Omega)((A, B)\in\mathscr{S} \ \& \ C \subseteq A \ \& \ D \subseteq B \to (C, D)\in\mathscr{S})$;

(S4) $(\forall A, B, C, D \subseteq \Omega)((A, B)\in\mathscr{S} \ \& \ (C, D)\in\mathscr{S} \to (A\cap C, B\cup D)\in\mathscr{S})$;

(S5) $(\forall A, B, C, D \subseteq \Omega)((A, B)\in\mathscr{S} \ \& \ (C, D)\in\mathscr{S} \to (A\cup C, B\cap D)\in\mathscr{S})$.

Először a teljes *f*- és a teljes *d*-családok közt fennálló dualitást fogalmazzuk meg pontosan a f- és a ϑ-axiómák alapján:

2.1. LEMMA: Legyen $\mathcal{F} \subseteq P(\Omega) \times P(\Omega)$ olyan halmaz, amelyre $(A, B) \in \mathcal{F}$ esetén $A \neq \emptyset$, ha $B \neq \emptyset$. Ekkor: $\mathcal{F}$-re igazak az f-axiómák akkor és csak akkor, ha $\mathcal{D} = \{(B, A) : (A, B) \in \mathcal{F}\}$-re igazak a ϑ-axiómák.

Bizonyítás: A bizonyítás nyilvánvaló az axiómákból. Jegyezzük meg, hogy az $(A, B) \in \mathcal{F} \to A \neq \emptyset$, ha $B \neq \emptyset$ feltétel (D5) miatt szükséges.

Megjegyzések. 1. Tegyük fel, hogy $\mathcal{F} \subseteq P(\Omega) \times P(\Omega)$ kielégíti az f-axiómákat és legyen $\mathcal{F}' = \mathcal{F} \setminus \{(\emptyset, B) : B \neq \emptyset\}$. Ekkor $\mathcal{F}'$-re teljesülnek a 2.1. lemma feltételei; és ha $A \subseteq \Omega$, $A \neq \emptyset$, akkor $(\forall B \subseteq \Omega)((A, B) \in \mathcal{F} \leftrightarrow (A, B) \in \mathcal{F}')$. Ez mutatja a 2.1 lemma feltételének technikai jellegét.

2. Jegyezzük még meg, hogy ha egy $\mathcal{F} \subseteq P(\Omega) \times P(\Omega)$ kielégíti az f-axiómákat, akkor a 2.1 lemma és az 1. megjegyzés alapján $\mathcal{F}$-hez a

$$\mathcal{D}(\mathcal{F}) = \{(B, A) : (A, B) \in \mathcal{F} \ \& \ B \neq \emptyset \to A \neq \emptyset\}$$

definícióval rendelt $\mathcal{D}(\mathcal{F})$ halmaz kielégíti a ϑ-axiómákat. Így persze különböző $\mathcal{F}$-ekhez rendelhettük ugyanazt a halmazt.

Most leírjuk a teljes *f*-, *d*- és *s*-családokat jellemző axiómákat; valamint ezek analogonját gyenge függésekre. A 3.3 tételben majd bebizonyítjuk, hogy ez utóbbi axióma nem jellemzi a teljes *w*-családokat.

F-axióma: $(\forall (X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathcal{F}) \exists E \subseteq \Omega$ úgy, hogy

(i) $X \subseteq E, \quad Y \not\subseteq E$

(ii) $(A, B) \in \mathcal{F} \quad \& \quad A \subseteq E \to B \subseteq E.$

D-axióma: $(\forall (X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathcal{D}) \exists E \subseteq \Omega$ úgy, hogy

(i) $X \cap E \neq \emptyset, \quad Y \cap E = \emptyset$

(ii) $(A, B) \in \mathcal{D} \quad \& \quad A \cap E \neq \emptyset \to B \cap E \neq \emptyset.$

S-axióma: $(\forall (X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathcal{S}) \exists E \subseteq \Omega$ úgy, hogy

(i) $X \cap E \neq \emptyset, \quad Y \not\subseteq E$

(ii) $(A, B) \in \mathcal{S} \quad \& \quad A \cap E \neq \emptyset \to B \subseteq E.$

W-axióma: $(\forall (X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathcal{W}) \exists E \subseteq \Omega$ úgy, hogy

(i) $X \subseteq E, \quad Y \cap E = \emptyset$

(ii) $(A, B) \in \mathcal{W} \quad \& \quad A \subseteq E \to B \cap E \neq \emptyset.$

2.1. TÉTEL: (a) Legyen $\mathcal{F} \subseteq P(\Omega) \times P(\Omega)$.
$\mathcal{F}$ kielégíti az f-axiómákat akkor és csak akkor, ha $\mathcal{F}$ kielégíti az *F*-axiómát.

(b) Legyen $\mathcal{D} \subseteq P(\Omega) \times P(\Omega)$.
$\mathcal{D}$ kielégíti a ϑ-axiómákat akkor és csak akkor, ha $\mathcal{D}$ kielégíti a *D*-axiómát.

(c) Legyen $\mathcal{S} \subseteq P(\Omega) \times P(\Omega)$.
$\mathcal{S}$ kielégíti a γ-axiómákat akkor és csak akkor, ha $\mathcal{S}$ kielégíti az *S*-axiómát.

Bizonyítás: (a) Tegyük fel először, hogy $\mathcal{F}$ kielégíti az f-axiómákat. Bebizonyítjuk, hogy ekkor $\mathcal{F}$ kielégíti az F-axiómát.

Legyen $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathcal{F}$ tetszőleges.

Állítjuk, hogy ekkor

(1) létezik olyan $E \subseteq \Omega$, amelyre $X \subseteq E$; $(E, Y) \in P(\Omega) \times P(\Omega) \setminus \mathcal{F}$, továbbá, ha $E' \supset E$, akkor $(E', Y) \in \mathcal{F}$ (azaz létezik maximális olyan X-et tartalmazó része Ω-nak, amelytől Y nem függ).

Ez világos, hiszen $(\Omega, \Omega) \in \mathcal{F}$ (F1) szerint és így (F3) miatt $(\Omega, Y) \in \mathcal{F}$, míg $(X, Y) \notin \mathcal{F}$.

Megmutatjuk, hogy az (1) szerinti E teljesíti az F-axióma (i) és (ii) feltételeit. (i) nyilvánvaló, mert $E \supseteq X$, E választása szerint és ha $E \supseteq Y$ állna fenn, akkor (F1) és (F3) miatt $(E, Y) \in \mathcal{F}$ lenne.

Az (ii) bizonyításához válasszunk egy tetszőleges olyan $(A, B) \in \mathcal{F}$-et, amelyre $A \subseteq E$. Tegyük fel indirekt, hogy $B \nsubseteq E$. Ekkor $E' := E \cup B \supset E$ és így $(E' Y) \in \mathcal{F}$ (1) szerint. De ekkor

$(E, E) \in \mathcal{F}$ (F1) miatt; így

$(E, E') \in \mathcal{F}$, mert $E = E \cup A$ és $E' = E \cup B$ és (F4); végül

$(E, Y) \in \mathcal{F}$ mivel $(E, E'), (E', Y) \in \mathcal{F}$ és (F2).

Ellentmondásra jutottunk, hiszen $(E, Y) \in \mathcal{F}$ lehetetlen (1) szerint.

A fordított irány könnyű. Példaként bebizonyítjuk, hogy ha $\mathcal{F}$-re igaz az F-axióma, akkor $\mathcal{F}$-re igaz (F2). Tegyük fel indirekt, hogy $(A, B), (B, C) \in \mathcal{F}$ és $(A, C) \in P(\Omega) \times P(\Omega) \setminus \mathcal{F}$. Az F-axióma szerint létezik $E \subseteq \Omega$ úgy, hogy $A \subseteq E$, $C \subseteq E$ és $(A, B) \in \mathcal{F}$, $A \subseteq E \to B \subseteq E$; így $(B, C) \in \mathcal{F}$ és $B \subseteq E$ miatt $C \subseteq E$, ami ellentmondás.

(b) Ennek bizonyítását a 2.1 lemmát használva végezzük. Legyen $\mathcal{F} = \{(A, B) : (B, A) \in \mathcal{D}\}$. A 2.1 lemma miatt elég bizonyítanunk, hogy $\mathcal{F}$ akkor és csak akkor elégíti ki az F-axiómát, ha $\mathcal{D}$ kielégíti a D-axiómát.

Tegyük fel, először, hogy $\mathcal{F}$-re igaz az F-axióma; $(A, B) \notin \mathcal{F}$ esetén legyen $E(A, B) \subseteq \Omega$ egy olyan, ami teljesíti az F-axióma feltételeit. Könnyű végiggondolni, hogy ekkor $E := \Omega \setminus E(A, B)$ kielégíti a D-axióma (i) és (ii) feltételeit (B, A)-ra és $\mathcal{D}$-re.

Ugyanúgy látható, hogy ha $\mathcal{D}$-re igaz a D-axióma, akkor $\mathcal{F}$-re igaz az F-axióma.

(c) Tegyük fel először, hogy $\mathcal{S}$-re igazak a γ-axiómák. Bebizonyítjuk, hogy ekkor $\mathcal{S}$ kielégíti az S-axiómát. Legyen $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathcal{S}$. Ekkor

(2) létezik $a \in X$ és $E \subseteq \Omega$ úgy, hogy $a \in E$, $(\{a\}, E) \in \mathcal{S}$ és ha $E' \supset E$, akkor $(\{a\}, E') \in P(\Omega) \times P(\Omega) \setminus \mathcal{S}$. Először is létezik $a \in X$, amelyre $(\{a\}, Y) \in P(\Omega) \times P(\Omega) \setminus \mathcal{S}$, mert ellenkező esetben (S5) ismételt alkalmazása $(X, Y) \in \mathcal{S}$-et eredményezné. Legyen tehát $a \in X$ olyan, hogy $(\{a\}, Y) \in P(\Omega) \times P(\Omega) \setminus \mathcal{S}$. Ekkor, (S4)-használva, létezik $b \in Y$, hogy $(\{a\}, \{b\}) \in P(\Omega) \times P(\Omega) \setminus \mathcal{S}$.

Végül (S1) és (S3) miatt létezik olyan $E \subseteq \Omega$, hogy $a \in E$, $(\{a\}, E) \in \mathcal{S}$ és ha $E' \supset E$, akkor $(\{a\}, E') \notin \mathcal{S}$. Ezzel (2)-t bebizonyítottuk.

Állítjuk, hogy E teljesíti az S-axióma feltételeit (X, Y)-ra.

(i): $a \in E \to X \cap E \neq \emptyset$ és $(\{a\}, \{b\}) \notin \mathcal{S}$ miatt (S3)-ból következik, hogy $Y \nsubseteq E$.

(ii): legyen $(A, B)\in\mathscr{S}$ olyan, hogy $A\cap E\neq\emptyset$. Tegyük fel indirekt, hogy $B\nsubseteq E$. Legyen $c\in A\cap E$ és $d\in B\cap(\Omega\setminus E)$. Ekkor $(\{c\}, \{d\})\in\mathscr{S}$ (S3) miatt; így $(\{a, c\}, \{c\})\in\mathscr{S}$ mert $(\{c\}, \{c\})\in\mathscr{S}$ (S1) szerint; $(\{a\}, E)\in\mathscr{S}$ és (S5); ezért $(\{a\}, \{c\})\in\mathscr{S}$ (S3) alapján; tehát $(\{a\}, \{d\})\in\mathscr{S}$ (S2) miatt. Végül is $(\{a\}, E\cup\{d\})\in\mathscr{S}$ (S4) szerint; ami ellentmondás, mert $E':=E\cup\{d\}\supset E$.

Ha $\mathscr{S}$ kielégíti az S-axiómát, akkor könnyen végiggondolható, hogy $\mathscr{S}$-re igazak a γ-axiómák.

3. Az egyenlőséghalmaz; a teljes w-családok axiomatizálása

Az utolsó paragrafust mátrixok egyenlőséghalmazának definíciójával kezdjük, majd ezek egyszerű jellemzését adjuk meg. E jellemzésre támaszkodva megfogalmazzuk az F'-, D'-, S'- és W'-axiómákat. Ezen axiómák lényegében nem mások, mint a megfelelő függések definícióinak egyszerű átfogalmazásai az egyenlőséghalmazokra. Bebizonyítjuk, hogy a vesszős axiómák rendre ekvivalensek vesszőtlen megfelelőikkel, kivéve a W-, ill. W'-axiómát (3.2 tétel).

Végül bebizonyítjuk, hogy az F'-, D'-, S'- és W'-axiómák rendre jellemzik a teljes f-, d-, s- és w-családokat.

3.1. Definíció: Legyen R reláció Ω felett és legyenek h, g R két sora. Definiáljuk h és g egyenlőséghalmazát, $E(h, g)$-t így:

$$E(h, g) = \{a\in\Omega:\ h(a) = g(a)\}.$$

Az $\{E(h, g): h, g\in R\ \&\ h\neq g\}$ halmazt az R *reláció egyenlőséghalmaz*ának nevezzük és ε_R-rel jelöljük. A relációk egyenlőséghalmazának jellemzéséhez szükséges a következő definíció.

3.2. Definíció: Legyen $\mathscr{A}$ egy halmazrendszer.

Azt mondjuk, hogy $\mathscr{A}$ *Δ-rendszer,* ha $A, B, C, D\in\mathscr{A}$, $A\neq B$, $C\neq D$ esetén $A\cap B=C\cap D$.

Megjegyzés: Könnyen látható, hogy egy $\mathscr{A}$ halmazrendszer akkor és csak akkor Δ-rendszer, ha $A, B\in\mathscr{A}$, $A\neq B$ esetén $A\cap B=\bigcap\mathscr{A}$

3.1. Tétel: (a) Legyen R reláció Ω felett és legyenek h, f, g sorai R-nek. Ekkor az $\{E(f, g), E(h, g), E(f, h)\}$ halmazrendszer Δ-rendszer.

(b) Legyen $\varepsilon=\{E_{i,j}: 1\leqq i<j\leqq k\}$ olyan halmazrendszer Ω részhalmazaiból, amelyre igaz a következő:
ha $1\leqq i<j<l\leqq k$, akkor az $\{E_{i,j}; E_{i,l}; E_{j,l}\}$ halmazrendszer Δ-rendszer. Ekkor létezik R reláció Ω felett, amelynek ε az egyenlőséghalmaza; azaz $\varepsilon=\varepsilon_R$.

Bizonyítás: (a) Szimmetria okokból nyilván elég bizonyítani, hogy

$$a\in E(h, g)\cap E(g, f)\rightarrow a\in E(h, f).$$

Ez azonban világos, hiszen

$$a \in E(h, g) \cap E(g, f) \leftrightarrow h(a) = g(a) = f(a) \rightarrow$$
$$\rightarrow h(a) = f(a) \leftrightarrow a \in E(h, f).$$

(b) Definiálunk egy relációt Ω felett, amelynek k sora van: $h_1, h_2, \ldots, h_k$. Legyen $h_1(a)=0$, ha $a \in \Omega$. Tegyük fel, hogy egy $n<k$-ra $h_1, \ldots, h_n$ már definiáltak úgy, hogy $1 \leqq i < j \leqq n$ esetén $E(h_i, h_j)=E_{i,j}$. Legyen $a \in \Omega$ esetén

$$h_{n+1}(a) = \begin{cases} h_i(a), & \text{ha} \quad i \leqq n \text{ és } a \in E_{i,n+1}; \\ \max\{h_i(b): 1 \leqq i \leqq n,\ b \in \Omega\}+1, & \text{ha} \quad a \notin \bigcup_{i=1}^{n} E_{i,n+1}. \end{cases}$$

Ekkor

(a) h_{n+1} jól definiált. Ehhez azt kell bizonyítani, hogy $a \in E_{i,n+1} \cap E_{j,n+1}$ esetén $h_i(a)=h_j(a)$. De $a \in E_{i,n+1} \cap E_{j,n+1} \rightarrow a \in E_{i,j}$, mert $\{E_{i,j}; E_{i,n+1}; E_{j,n+1}\}$ Δ-rendszer a feltevés szerint és így $E_{i,j}=E(h_i, h_j)$ miatt $h_i(a)=h_j(a)$.

(b) ha $1 \leqq i \leqq n$ és $a \notin E_{i,n+1}$ akkor $h_i(a) \neq h_{n+1}(a)$. Ha $a \notin \bigcup_{i=1}^{n} E_{i,n+1}$, akkor h_{n+1} definíciója szerint $h_i(a) \neq h_{n+1}(a)$. Ha $a \in E_{j,n+1}$, akkor $h_j(a)=h_{n+1}(a)$ és $h_j(a) \neq h_i(a)$, mert $\{E_{i,j}; E_{i,n+1}; E_{j,n+1}\}$ Δ-rendszer volta miatt $a \notin E_{i,j}=E(h_i, h_j)$.

(a)-ból és (b)-ből könnyen következik, hogy $E_{i,n+1}=E(h_i, h_{n+1})$, ha $1 \leqq i \leqq n$. Tehát $\{h_1, \ldots, h_k\}=R$-re $\varepsilon=\varepsilon_R$.

A különböző függések és a sorok egyenlőséghalmazának definíciói alapján nem nehéz látni, hogy a következő axiómák a függések definícióinak átfogalmazásai egyenlőséghalmazokra.

Az axiómák (iii) feltétele a 3.1 tétel miatt szükséges.

F'-axióma: $\exists\, \{E_{i,j}: 1 \leqq i < j \leqq k\} \subseteq P(\Omega)$ úgy, hogy

(i) $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathscr{F} \rightarrow (\exists i, j)(X \subseteq E_{i,j} \ \&\ Y \not\subseteq E_{i,j})$

(ii) $(A, B) \in \mathscr{F}, \quad 1 \leqq i < j \leqq k, \quad A \subseteq E_{i,j} \rightarrow B \subseteq E_{i,j}$

(iii) $(\forall\, 1 \leqq i < j < l \leqq k)(\{E_{i,j}; E_{i,l}; E_{j,l}\}$ Δ-rendszer).

D'-axióma: $\exists \{E_{i,j}: 1 \leqq i < j \leqq k\} \subseteq P(\Omega)$ úgy, hogy

(i) $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathscr{D} \rightarrow (\exists i, j)(X \cap E_{i,j} \neq \emptyset \ \&\ Y \cap E_{i,j} = \emptyset)$

(ii) $(A, B) \in \mathscr{D}, \quad 1 \leqq i < j \leqq k, \quad A \cap E_{i,j} \neq \emptyset \rightarrow B \cap E_{i,j} \neq \emptyset$

(iii) $(\forall\, 1 \leqq i < j < l \leqq k)(\{E_{i,j}; E_{i,l}; E_{j,l}\}$ Δ-rendszer).

S'-axióma: $\exists \{E_{i,j}: 1 \leqq i < j \leqq k\} \subseteq P(\Omega)$ úgy, hogy

(i) $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathscr{S} \rightarrow (\exists i, j)(X \cap E_{i,j} \neq \emptyset \ \&\ Y \not\subseteq E_{i,j})$

(ii) $(A, B) \in \mathscr{S}, \quad 1 \leqq i < j \leqq k, \quad A \cap E_{i,j} \neq \emptyset \rightarrow B \subseteq E_{i,j}$

(iii) $(\forall\, 1 \leqq i < j < l \leqq k)(\{E_{i,j}; E_{i,l}; E_{j,l}\}$ Δ-rendszer).

W'-axióma: $\exists\{E_{i,j}: 1 \leqq i < j \leqq k\} \subseteq P(\Omega)$ úgy, hogy

(i) $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathscr{W} \to (\exists i, j)(X \subseteq E_{i,j}) \ \& \ Y \cap E_{i,j} = \emptyset)$

(ii) $(A, B) \in \mathscr{W}, \quad 1 \leqq i < j \leqq k, \quad A \in E_{i,j} \to B \cap E_{i,j} \neq \emptyset$

(iii) $(\forall\, 1 \leqq i < j < l \leqq k)(\{E_{i,j}; E_{i,l}; E_{j,l}\}$ Δ-rendszer).

3.2. TÉTEL (a) Legyen $\mathscr{Y} \subseteq P(\Omega) \times P(\Omega)$.

$\mathscr{Y}$ kielégíti az F-axiómát akkor és csak akkor, ha $\mathscr{Y}$ kielégíti az F'-axiómát.
$\mathscr{Y}$ kielégíti a D-axiómát akkor és csak akkor, ha $\mathscr{Y}$ kielégíti a D'-axiómát.
$\mathscr{Y}$ kielégíti az S-axiómát akkor és csak akkor, ha $\mathscr{Y}$ kielégíti az S'-axiómát.

(b) Legyen Ω véges halmaz, $|\Omega| \geqq 3$. Ekkor létezik olyan $\mathscr{W} \subseteq P(\Omega) \times P(\Omega)$, amely kielégíti a W-axiómát, de nem elégíti ki a W'-axiómát.

Bizonyítás: (a) Tegyük fel, hogy $\mathscr{Y}$ kielégíti az F-axiómát. $(X, Y) \in P(\Omega) \times$ $\times P(\Omega) \setminus \mathscr{Y}$-re legyen $E(X, Y) \subseteq \Omega$ az F-axióma szerint (X, Y)-hoz létező és legyen $\{E_2, \ldots, E_k\} = \{E(X, Y): (X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathscr{Y}\}$. Definiáljuk az $\{E_{i,j}: 1 \leqq i < j \leqq k\}$ halmazrendszert így:

ha $1 < j \leqq k$, akkor legyen $E_{1,j} = E_j$ és ha $1 < i < j \leqq k$, akkor legyen $E_{i,j} = E_i \cap E_j$.

Világos, hogy ekkor $\{E_{i,j}: 1 \leqq i < j \leqq k\}$ teljesíti az F'-axióma (i) és (ii) feltételét $\mathscr{Y}$-nal. (iii) bizonyításához legyen $1 \leqq i < j < l \leqq k$.

Ha $i = 1$ akkor $E_{i,j} = E_j$, $E_{i,l} = E_l$, $E_{j,l} = E_j \cap E_l$, tehát $\{E_{i,j}; E_{i,l}; E_{j,l}\}$ bármely két elemének metszete $E_i \cap E_j$ és így ez Δ-rendszer. Ha $i > 1$, akkor $E_{i,j} =$ $= E_i \cap E_j$; $E_{i,l} = E_i \cap E_l$ és $E_{j,l} = E_j \cap E_l$ és így $\{E_{i,j}; E_{i,l}; E_{j,l}\}$ bármely két elemének metszete $E_i \cap E_j \cap E_l$, tehát Δ-rendszer. Ha $\mathscr{Y}$ kielégíti az F'-axiómát, akkor $\mathscr{Y}$ kielégíti az F-axiómát — ez nyilvánvaló.

Tegyük fel most, hogy $\mathscr{Y}$ kielégíti a D-axiómát. $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathscr{Y}$-ra legyen $E(X, Y) \subseteq \Omega$ a D-axióma szerint (X, Y)-hoz létező és legyen

$$\{E_1, \ldots, E_k\} = \{E(X, Y): (X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathscr{Y}\}.$$

Definiáljuk az $\{E_{i,j}: 1 \leqq i < j \leqq 2k\}$ halmazrendszert a következőképpen:
$1 \leqq i \leqq k$-ra legyen $E_{2i-1,2i} = E_i$; ha $1 \leqq i < j \leqq 2k$ és $E_{i,j}$ még nem definiált, akkor pedig legyen $E_{i,j} = \emptyset$.

Ekkor $\{E_{i,j}: 1 \leqq i < j \leqq 2k\}$ kielégíti a D'-axióma (i) és (ii) feltételét $\mathscr{Y}$-nal; ez világos. (iii) azért igaz, mert ha $1 \leqq i < j < l \leqq 2k$, akkor $\{E_{i,j}; E_{i,l}; E_{j,l}\}$ legalább két eleme $\emptyset$, tehát Δ-rendszer.

Ha $\mathscr{Y}$ kielégíti a D'-axiómát, akkor nyilván kielégíti a D-axiómát.

Az olvasóra bízzuk annak bizonyítását, hogy ha $\mathscr{Y}$ kielégíti az S-axiómát, akkor $\mathscr{Y}$ kielégíti az S'-axiómát is.

(b) Az egyszerűség kedvéért legyen $\Omega = \{a, b, c\}$ (az általános esetben $\{c\}$ szerepét $\Omega \setminus \{a, b\}$ játssza).

Legyen

$$\mathscr{W} = \{(A, B) \in P(\Omega) \times P(\Omega): A \subseteq \{a\} \to a \in B \ \& \ A \subseteq \{b\} \to b \in B\}.$$

$\mathscr{W}$ kielégíti a W-axiómát, mert ha $(X, Y)\in P(\Omega)\times P(\Omega)\setminus\mathscr{W}$, akkor vagy $X\subseteq\{a\}$ és $a\notin Y$, vagy $X\subseteq\{b\}$ és $b\notin Y$; az első esetben $E=\{a\}$, a másodikban $E=\{b\}$ mutatja, hogy igaz a W-axióma.

Tegyük fel indirekt, hogy $\varepsilon=\{E_{i,j}: 1\leqq i<j\leqq k\}$ mutatja a W'-axióma teljesülését $\mathscr{W}$-re. Ekkor

(1) $\{a\}\in\varepsilon$ és $\{b\}\in\varepsilon$, mert $(\{a\}, \Omega\setminus\{a\})\notin\mathscr{W}$ és $(\{b\}, \Omega\setminus\{b\})\notin\mathscr{W}$.

(2) $\emptyset\notin\varepsilon$ és $\{c\}\notin\varepsilon$, mert $(\emptyset, \Omega)\in\mathscr{W}$ és $(\{c\}, \Omega\setminus\{c\})\in\mathscr{W}$.

$\{a\}$ és $\{b\}$ ε-beliek; „elhelyezkedésüket" tekintve két esetet különböztetünk meg:

(a) $\{a\}=E_{i,j}$ és $\{b\}=E_{i,l}$. Ekkor; mivel $\{E_{i,j}; E_{j,l}; E_{i,l}\}$-rendszer; $E_{j,l}$ csak $\emptyset$ vagy $\{c\}$ lehet, ami ellentmond (2)-nek.

(b) $\{a\}=E_{i,j}$ és $\{b\}=E_{l,m}$, ahol $|\{i, j, l, m\}|=4$.

Mivel a bizonyításban csak $\{E_{i,j}; E_{i,l}; E_{i,m}; E_{j,l}; E_{j,m}; E_{l,m}\}$-mel foglalkozunk, feltehető $i=1, j=2, l=3, m=4$. Azt vizsgáljuk, hogy $E_{1,3}$ mi lehet. Az $E_{1,3}=\{a\}$ és az $E_{1,3}=\{b\}$ esetek az (a) esethez vezetnek.

$E_{1,3}\neq\{c\}$ és $E_{1,3}\neq\emptyset$ (2) miatt. $E_{1,3}\neq\{b, c\}$ mert $\{E_{1,2}; E_{1,3}; E_{2,3}\}$ Δ-rendszer, és ha $E_{1,3}=\{b, c\}$, akkor így $E_{2,3}=\emptyset$, ami ellentmond (2)-nek. Tehát $a\in E_{1,3}$. Így $a\in E_{2,3}$, mert $\{E_{1,2}; E_{1,3}; E_{2,3}\}$ Δ-rendszer. $a\notin E_{2,4}$, mert $\{E_{2,3}; E_{2,4}; E_{3,4}\}$ Δ-rendszer. Így $E_{2,4}\subseteq\{b, c\}$. $E_{2,4}\neq\{c\}$ és $E_{2,4}\neq\emptyset$ (2) miatt és $E_{2,4}\neq\{b\}$ az (a) eset miatt. Tehát $E_{2,4}=\{b, c\}$.

Így $b\in E_{2,3}$, mert $\{E_{2,3}; E_{2,4}; E_{3,4}\}$ Δ-rendszer. Végül így $E_{1,3}=\{a, c\}$, mert $\{E_{2,3}; E_{1,2}; E_{1,3}\}$ Δ-rendszer. $\{E_{1,3}; E_{1,4}; E_{3,4}\}$ Δ-rendszer; $E_{1,3}\cap E_{3,4}=\emptyset$, $E_{1,3}\cup\cup E_{3,4}=\Omega$, ezért $E_{1,4}=\emptyset$, ami ellentmond (2)-nek.

Megjegyzés: A 3.2 tétel lényeges különbséget mutat a gyenge és a többi függés között. Ez a különbség lényegében az, hogy az $\emptyset$ kezdőtagú függések csak a gyenge függésnél bírnak más tulajdonságokkal, mint a nem $\emptyset$ kezdőtagúak. Jegyezzük még meg, hogy az F'-axiómában szereplő $E_{i,j}$-k (és persze az F-axiómabeli E-k) maximális jobb oldalak; ennek alapján analóg módon definiálható a többi függésre is a maximális függés fogalma és ARMSTRONG erre vonatkozó eredményei (a maximális jobb oldalak metszetre zártságát kivéve) nehézség nélkül adaptálhatók. Végül bebizonyítjuk, hogy axiómáink valóban jellemzik a teljes családokat.

3.3. TÉTEL: Legyen $\mathscr{Y}\subseteq P(\Omega)\times P(\Omega)$. Ekkor $\mathscr{Y}$ kielégíti az F'- (D'-, S'-, W'-) axiómát akkor és csak akkor, ha létezik olyan R reláció Ω felett, amelyre $\mathscr{Y}=\mathscr{F}_R(\mathscr{D}_R, \mathscr{S}_R, \mathscr{W}_R)$.

Bizonyítás: Tegyük fel először, hogy $\mathscr{Y}$ kielégíti az Y'-axiómát valamely $Y\in\{F, D, S, W\}$-re. Ekkor az Y'-axióma (iii) feltétele és 3.1 tétel (b) pontja szerint létezik olyan R reláció, amelynek egyenlőséghalmaza az Y'-axióma által garantált halmazr endszer. R egyenlőséghalmazának definíciója és az Y'-axióma (i) és (ii) feltételei m iatt nyilvánvaló, hogy ekkor $\mathscr{Y}$ az R megfelelő típusú függéseinek rendszere.

For ditva, ha R egy reláció Ω felett, $R=\{h_1, \ldots, k_k\}$, akkor $\varepsilon_R=\{E(h_i, h_j): 1\leqq i<j\leqq k\}$ mutatja, hogy R f-, d-, s-, ill. w-függéseinek halmazai kielégítik az F'-, D'-S'-, ill. W'-axiómát.

IRODALOM

[1] ARMSTRONG, W. W., "Dependency structures of data base relationship", *Information Processing 74*, North-Holland Publ. Co. 1974, 580—583.
[2] ARMSTRONG, W. W., "On the generation of dependency structures of relational data bases", Publication 272, Université de Montréal, 1977.
[3] ARMSTRONG, W. W. and DELOBEL, C., "Decomposition and functional dependencies in relations", Publication 271, Université de Montreal, 1979.
[4] BEERI, C., FAGIN, R. and HOWARD, J. H., "A complete axiomatization for functional and multivalued dependencies in database relations", *Proc. ACM SIGMOND Int. Conf. an Maganement of Data,* Toronto 1977, 47—61.
[5] BÉKÉSSY, A. and DEMETROVICS, J., "Contribution to the theory of data base relations", *Discrete Math.* **27** (1979) 1—10.
[6] CODD, E. F., "A relational model of data for large shared data banks", *Comm. ACM*, **13** (1970) 377—387.
[7] CODD, E. F., "Further normalization of the data base relational model", *Courant Computer Science Symposia* **6** *Data Base System,* Prentice Hall, Englewood Cliffs, N. J. 1971, 33—64.
[8] CZÉDLI, G., „Függőségek relációs adatbázis modellben", *Alk. Mat. Lapok*, **6** (1980) 131—144.
[9] DEMETROVICS, J., "Candidate keys and antichains", *SIAM J. on Algebraic and Discrete Methods* **1** (1980) 1—92.
[10] DEMETROVICS, J. GYEPESI, GY., "Some general dependencies in the relational data model", *Acta Cybernetica* **5** (1981) 295—305.

(Beérkezett: 1980. május 17.)

DEMETROVICS JÁNOS ÉS GYEPESI GYÖRGY
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1132 BUDAPEST, VICTOR HUGO U. 18.

GENERALIZED FUNCTIONAL DEPENDENCIES IN RELATIONAL DATA BASES

J. DEMETROVICS and GY. GYEPESI

In this paper we deal with the functional dependencies of the relational data model and their three generalizations: the dual, strong and weak dependencies. We axiomatize the full families of these dependencies of any kind (full families of functional dependencies were firstly axiomatized in [1] and that of dual and strong dependencies in [8] as well; the axiomatization of full families of weak dependencies is a new result).

Our axioms are based on a combinatorial characterization of equality-sets of matrices. We prove an important difference between the weak dependencies and the rest.

Alkalmazott Matematikai Lapok 6 (1980) 323—329

FUNKCIONÁLIS FÜGGŐSÉGEK TELJES CSALÁDJAINAK GENERÁLÁSA ÉS RELÁCIÓKKAL VALÓ REPREZENTÁLÁSA

DEMETROVICS JÁNOS ÉS GYEPESI GYÖRGY

Budapest

Ebben a dolgozatban a funkcionális függőség elméletének két kombinatorikus problémájával foglalkozunk. Ez a két probléma a teljes családok tárolási problémájával kapcsolatos.

Ha R reláció az n elemű A halmaz felett, akkor az R által teljesített funkcionális függőségek száma 3^n és 4^n között van.

Jelölje $S(n)$ azt a legkisebb egész számot, amelyre minden $F \subseteq P(A) \times P(A)$ teljes család egy legfeljebb $S(n)$ sorral rendelkező A halmaz feletti alkalmas reláció funkcionális függőségeinek a halmaza. Bebizonyítjuk, hogy $1/n^2 \binom{n}{[n/2]} \leqq S(n) \leqq 3/2 \binom{n}{[n/2]}$. Ezért gazdaságos egy adott teljes családot a megfelelő relációk segítségével tárolni.

Az $S(n)$ szám egy változata $s(n)$, ahol $s(n)$ az a legkisebb egész szám amelyre minden $C \subseteq P(A)$ *Sperner rendszer* egy legfeljebb $s(n)$ sorral rendelkező A halmaz feletti alkalmas reláció kijelölt kulcsainak a halmaza. Bebizonyítjuk, hogy $1/n^2 \binom{n}{[n/2]} \leqq s(n) \leqq \binom{n}{[n/2]} + 1$.

Végül a teljes családok minimális számosságú generátorrendszereire adunk kombinatorikus jellemzést. Bebizonyítjuk, hogy ez a minimális számosság kisebb mint $3/2 \binom{n}{[n/2]}$.

1. Bevezetés

Az E. F. CODD [5] által bevezetett relációs adatmodell az adatok tárolását mátrixok alakjában valósítja meg. A mátrixok sorai az adatrekordok, az oszlopok pedig az ún. attributumok, azaz azok a tulajdonságok, melyekre az egyes adatok vonatkoznak.

A pontos definíció a következő:

Legyen $\Omega = \{a_1, \ldots, a_n\}$ egy véges nem üres halmaz. Ekkor egy R véges halmaz egy *Ω feletti reláció,* ha R elemei Ω-n értelmezett függvények. Az R-et alkotó függvényeket *R sorainak* nevezzük.

Egy $R = \{f_1, \ldots, f_k\}$ Ω feletti relációt kétdimenziós táblázat formájában képzelhetünk el:

	a_1	a_2	...	a_n
f_1	$f_1(a_1)$	$f_1(a_2)$	...	$f_1(a_n)$
f_2	$f_2(a_1)$	$f_2(a_2)$	...	$f_2(a_n)$
⋮	⋮	⋮		⋮
f_k	$f_k(a_1)$	$f_k(a_2)$	...	$f_k(a_n)$

A relációs adatbázisok információszolgáltatásával áll szoros kapcsolatban a *funkcionális függés* [6] fogalma.

Legyen R egy Ω feletti reláció, és legyenek A, B részhalmazai Ω-nak. Azt mondjuk, hogy B *funkcionálisan függ A-tól R-ben* $\left(A \xrightarrow[R]{f} B\right)$, ha R bármely két f, g sorára igaz, hogy

$$(\forall a \in A)(f(a) = g(a)) \rightarrow (\forall b \in B)(f(b) = g(b)),$$

azaz ha R-ben minden sor B-beli értékeit egyértelműen meghatározzák a sor A-beli értékei. Jelölje $\mathscr{F}_R$ a továbbiakban az R reláció funkcionális függőségeinek halmazát, azaz

$$\mathscr{F}_R = \{(A, B): A \xrightarrow[R]{f} B\}.$$

Így $\mathscr{F}_R \subseteq P(\Omega) \times P(\Omega)$. Nevezzük teljes családoknak azon $\mathscr{Y}$ részhalmazait a $P(\Omega) \times$ $\times P(\Omega)$-nak, melyek valamely Ω feletti reláció funkcionális függőségei; azaz egy $\mathscr{Y} \subseteq P(\Omega) \times P(\Omega)$ teljes család akkor és csak akkor, ha létezik olyan R reláció Ω felett, amelyre $\mathscr{Y} = \mathscr{F}_R$.

A funkcionális függőségek vizsgálata során az első feladat a teljes családok absztrakt jellemzése volt. Ez W. W. ARMSTRONGnak [1] sikerült először. Bebizonyította, hogy egy $\mathscr{Y} \subseteq P(\Omega) \times P(\Omega)$ akkor és csak akkor teljes család, ha $\mathscr{Y}$-ra teljesülnek a következők (A, B, C, D tetszőleges részei Ω-nak):

(F1) $(A, A) \in \mathscr{Y}$;

(F2) $(A, B) \in \mathscr{Y}$, $(B, C) \in \mathscr{Y} \rightarrow (A, C) \in \mathscr{Y}$;

(F3) $(A, B) \in \mathscr{Y}$, $C \supseteq A$, $D \subseteq B \rightarrow (C, D) \in \mathscr{Y}$;

(F4) $(A, B) \in \mathscr{Y}$, $(C, D) \in \mathscr{Y} \rightarrow (A \cup C, B \cup D) \in \mathscr{Y}$.

A funkcionális függések további vizsgálata során felmerült problémák a teljes családok kombinatorikus struktúrájával állnak szoros kapcsolatban (l. [3], [4], [7]).

Ebben a cikkben a teljes családok elméletének két problémájával foglalkozunk. E problémák a következő általános kérdéshez kapcsolódnak: általában mennyi „információ" elegendő egy teljes család leírásához.

A problémák precízen fogalmazva a következők:

1. PROBLÉMA. Legyen $|\Omega| = n$. Mi az a legkisebb $S(n)$ természetes szám, amelyre igaz a következő: ha $\mathscr{Y} \subseteq P(\Omega) \times P(\Omega)$ teljes család, akkor van olyan R reláció Ω felett, hogy $\mathscr{Y} = \mathscr{F}_R$ és $|R| \leqq S(n)$.

2. PROBLÉMA. Legyen $\mathscr{F} \subseteq P(\Omega) \times P(\Omega)$ teljes család. Jellemezzük $\mathscr{F}$ minimális számosságú generátorrendszereit.

A 2. pontban az 1. problémát, a 3. pontban pedig a 2. problémát tárgyaljuk.

2. Az 1. Probléma

Ebben a paragrafusban becslést adunk $S(n)$-re és $s(n)$-re (1.2.3 definíció). Az $s(n)$ becslésének problémáját [7] vetette fel, ahol $\sqrt{2 \cdot \binom{n}{[n/2]}} \leqq s(n) \leqq 2 \cdot \binom{n}{[n/2]}$ bizonyított. A 2.3 tétel ezt a becslést élesíti. A továbbiakban néhány definícióra lesz szükségünk.

2.1. DEFINÍCIÓ. Legyen $\mathcal{F} \subseteq P(\Omega) \times P(\Omega)$ egy teljes család, és legyen $(A, B) \in \mathcal{F}$. Ekkor *(A, B) maximális függése $\mathcal{F}$-nek,* ha $B' \supseteq B$, $(A, B') \in \mathcal{F} \to B' = B$. $M(\mathcal{F})$ jelöli $\mathcal{F}$ maximális függéseinek halmazát. Egy $B \subseteq \Omega$ *maximális jobb oldal $\mathcal{F}$-re,* ha létezik $A \subseteq \Omega$, amelyre $(A, B) \in M(\mathcal{F})$. $I(\mathcal{F})$ jelöli az $\mathcal{F}$-re maximális jobb oldalak halmazát.

2.1. LEMMA [1]. Legyen $\mathcal{F} \subseteq P(\Omega) \times P(\Omega)$. Ekkor

$$I(\mathcal{F}) = \{B \subseteq \Omega : (\forall (A, C) \in \mathcal{F})(A \subseteq B \to C \subseteq B)\}.$$

2.1. Lemma könnyű bizonyításának elvégzését az olvasóra bízzuk.

2.2. KÖVETKEZMÉNY. [1]. Legyen $\mathcal{F} \subseteq P(\Omega) \times P(\Omega)$ teljes család. Ekkor $I(\mathcal{F})$ metszetre zárt (azaz $B, B' \in I(\mathcal{F}) \to B \cap B' \in I(\mathcal{F})$).

Bizonyítás: 2.1 lemma szerint $I(\mathcal{F}) = \{B \subseteq \Omega : (A, C) \in \mathcal{F}, A \subseteq B \to C \subseteq B\}$. Legyenek B, B' tetszőleges elemei $I(\mathcal{F})$-nek; megmutatjuk, hogy $B \cap B' \in I(\mathcal{F})$. Ehhez azt kell bizonyítani, hogy $(A, C) \in \mathcal{F}$, $A \subseteq B \cap B'$ esetén $C \subseteq B \cap B'$. De ha $A \subseteq B \cap B'$, akkor $A \subseteq B$, $A \subseteq B'$, ezért $C \subseteq B$, $C \subseteq B'$ mert $B, B' \in I(\mathcal{F})$. Így $C \subseteq B \cap B'$ valóban.

2.2. DEFINÍCIÓ. Legyen $\mathcal{F} \subseteq P(\Omega) \times P(\Omega)$ egy teljes család. Egy $A \subseteq \Omega$ *kulcsa $\mathcal{F}$-nek,* ha $(A, \Omega) \in \mathcal{F}$; *$A$ kijelölt kulcsa $\mathcal{F}$-nek,* ha $(A, \Omega) \in \mathcal{F}$ és $A' \subseteq A$, $(A', \Omega) \in \mathcal{F} \to A' = A$. $K(\mathcal{F})$ jelöli az $\mathcal{F}$ kijelölt kulcsainak rendszerét.

Jegyezzük meg, hogy ha $\mathcal{F}$ teljes család, akkor $K(\mathcal{F})$ *Sperner-rendszer* [7] és így $|K(\mathcal{F})| \leq \binom{n}{[n/2]}$ ([9]).

2.3. DEFINÍCIÓ. Legyen $n > 0$ egy természetes szám és $|\Omega| = n$. Jelölje $s(n)$ azt a legkisebb természetes számot, amelyre: ha $K \subseteq P(\Omega)$ *Sperner-rendszer,* akkor van olyan R reláció Ω felett, amelyre $K = K(\mathcal{F}_R)$ és R sorainak száma legalább $s(n)$.

Megjegyzés: Ha $K \subseteq P(\Omega)$ *Sperner-rendszer,* akkor létezik olyan R reláció Ω felett, amelyre $K = K(\mathcal{F}_R)$; ez [7]-ben bizonyított, de a 2.3 tételből is következik.

2.3. TÉTEL.

$$\frac{1}{n^2}\binom{n}{[n/2]} \leq s(n) \leq \binom{n}{[n/2]} + 1.$$

Bizonyítás: (a) Először a felső korlát helyességét bizonyítjuk. Legyen $|\Omega| = n$ és $K \subseteq P(\Omega)$ egy tetszőleges *Sperner-rendszer*. Legyen L az Ω azon részhalmazainak rendszere, melyek K egyetlen elemét sem tartalmazzák és maximálisak erre a tulajdonságra, azaz

$$L = \{X \subseteq \Omega : (A \in K \to A \not\subseteq X) \ \& \ (Y \supsetneq X) \to (\exists A \in K)(A \subseteq Y)\}.$$

Világos, hogy L *Sperner-rendszer* és így $|L| \leq \binom{n}{[n/2]}$. Jelölje k az L számosságát és soroljuk fel L elemeit $X_1, \ldots, X_k$ alakban.

R Ω feletti relációt konstruálunk úgy, hogy R sorainak száma $k+1$ és $K(\mathcal{F}_R) = K$. Legyenek R sorai $(h_0, \ldots, h_k)$ a következők: $h_0(a) = 0$ minden $a \in \Omega$-ra, és ha i olyan, hogy $1 \leq i \leq k$, akkor $h_i(a) = \begin{cases} 0, & \text{ha } a \in X_i; \\ i, & \text{ha } a \in \Omega \setminus X_i. \end{cases}$ Legyen $R = \{h_0, h_1, \ldots, k_k\}$

Állítjuk, hogy $K(\mathscr{F}_R)=K$.

1. $A\in K\to A\in K(\mathscr{F}_R)$. Ugyanis ha $A\in K$, akkor R bármely két sora különbözik A-nak egy elemén, mivel $A\not\subseteq X_i$, ha $1\leqq i\leqq k$. Tehát A kulcsa $\mathscr{F}_R$-nek. A kijelölt kulcsa $\mathscr{F}_R$-nek, mert, ha $B\subsetneq A$ akkor K *Sperner-rendszer* volta miatt B nem tartalmazza K egyetlen elemét sem. Ezért létezik $i(1\leqq i\leqq k)$, hogy $B\subseteq X_i$. $K\subseteq P(\Omega)$ és L definíciója miatt $X_i\neq\Omega$, így $h_0\neq h_i$, de $B\subseteq X_i$ miatt $(\forall b\in B)(h_0(b)=h_i(b))$. Így B nem kulcsa $\mathscr{F}_R$-nek. Ezzel beláttuk, hogy $K\subseteq K(\mathscr{F}_R)$.

2. $K(\mathscr{F}_R)\subseteq K$.

Tegyük fel indirekt, hogy létezik $B\in K(\mathscr{F}_R)\setminus K$. Az 1. szerint $K\subseteq K(\mathscr{F}_R)$ és $K(\mathscr{F}_R)$ *Sperner-rendszer,* ezért B nem tartalmazza K egyetlen elemét sem. Így létezik i $(1\leqq i\leqq k)$, hogy $B\subseteq X_i$. Az 1. bizonyításában azonban láttuk, hogy ekkor B nem kulcsa $\mathscr{F}_R$-nek, ami ellentmondás.

Tehát $K(\mathscr{F}_R)\subseteq K$ és így végül $K=K(\mathscr{F}_R)$.

(b) Az alsó korlátra L. RÓNYAI bizonyítását közöljük:

Két könnyen látható észrevétellel kezdjük:

1. Ha R s-soros reláció, akkor létezik olyan R' s-soros reláció, amelyben legfeljebb s szimbólum fordul elő és amelyre $\mathscr{F}_R=\mathscr{F}_{R'}$ (így persze $K(\mathscr{F}_R)=K(\mathscr{F}_{R'})$).

2. Ha R s-soros reláció és $s'>s$ akkor létezik olyan R' s'-soros reláció, amelyre $\mathscr{F}_R=\mathscr{F}_{R'}$ (így persze $K(\mathscr{F}_R)=K(\mathscr{F}_{R'})$).

Az 1. és 2. alapján legfeljebb $s(n)$ soros relációk kijelölt kulcsinak rendszerei legfeljebb $s(n)^{s(n)\cdot n}$ sokan vannak. Másrészt n-elemű halmaznak több, mint $2^{\binom{n}{[n/2]}}$ *Sperner-rendszere* van, tehát

$$s(n)^{s(n)\cdot n}>2^{\binom{n}{[n/2]}},$$

azaz

$$s(n)\cdot n\cdot\log_2 s(n)>\binom{n}{[n/2]}.$$

Jelölje $a(n)$ az $\binom{n}{[n/2]}$-et. Legyen f olyan függvény, hogy

(1) $a(n)/n\geqq f(a(n))\cdot\log_2 f(a(n))$.

Ekkor $s(n)\geqq f(a(n))$, mert ha $s(n)<f(a(n))$, akkor $a(n)<n\cdot s(n)\cdot\log_2 s(n)<n\cdot f(a(n))\cdot\log_2 f(a(n))$, ami ellentmondás. Könnyű ellenőrizni, hogy ha $f(a(n))=a(n)/n^2$, akkor (1) teljesül f-re, tehát

$$s(n)\geqq\frac{1}{n^2}\binom{n}{[n/2]}.$$

Megjegyzés: A 2.3 tétel szerint $\frac{1}{n^2}\binom{n}{[n/2]}\leqq s(n)$, de nem tudunk konstruálni olyan *Sperner rendszert,* ami ezt mutatná.

2.4. KÖVETKEZMÉNY. $\frac{1}{n^2}\binom{n}{[n/2]}\leqq S(n)$.

2.5. TÉTEL. $S(n)\leqq\frac{3}{2}\binom{n}{[n/2]}+1$.

Bizonyítás: W. W. ARMSTRONG [1] bebizonyította, hogy ha $I\subseteq\Omega$ metszetre zárt, akkor pontosan egy olyan $\mathscr{F}$ teljes család van, amelyre $I=I(\mathscr{F})$. Ezért elég bizo-

nyítanunk, hogy ha $I \subseteq \Omega$ metszetre zárt, akkor létezik egy legfeljebb $\frac{3}{2}\binom{n}{[n/2]}+1$ soros R reláció, amelyre $I=I(\mathscr{F}_R)$. Legyen tehát $I \subseteq \Omega$ metszetre zárt. Legyen $N=\{Y \in I: Y \neq \cap\{Y' \in I: Y' \supseteq Y, Y' \neq Y\}\}$. Ekkor N olyan, hogy ha $Y \neq Y' \in N$, úgy $Y \cap Y' \notin N$. D. KLEITMANN [8] tétele szerint ekkor $|N|=k \leqq \frac{3}{2}\binom{n}{[n/2]}$.

A 3.1 lemmában bebizonyítjuk, hogy a legszűkebb olyan metszetre zárt rendszer, amely N-et tartalmazza, I. Ezért elegendő olyan R relációt konstruálnunk, melynek legfeljebb $\frac{3}{2}\binom{n}{[n/2]}+1$ sora van, és amelyre $N \subseteq I(\mathscr{F}_R) \subseteq I$. Legyen $N=\{Y_1, \ldots, Y_k\}$ és legyen R az a reláció, amelyet 2.3 tétel bizonyításában konstruáltunk; X_i helyére Y_i-t írva, ha $i=1, \ldots, k$. Nyilvánvaló, hogy ekkor $N \subseteq I(\mathscr{F}_R) \subseteq I$.

3. A 2. Probléma

Most rátérünk a 2. problémára, melyet W. W. ARMSTRONG [2] vizsgált először.

3.1. DEFINÍCIÓ. Legyen $\mathscr{F} \subseteq P(\Omega) \times P(\Omega)$ egy teljes család. Azt mondjuk, hogy egy $\mathscr{F}' \subseteq \mathscr{F}$ *generálja* $\mathscr{F}$*-et,* ha minden olyan R relációra Ω felett, amelyre $\mathscr{F}' \subseteq \mathscr{F}_R$, $\mathscr{F} \subseteq \mathscr{F}_R$ is igaz. A [2]-ben logikai jellemzése adott a teljes családok minimális számosságú generátorrendszereinek. Ezen jellemzés egy kombinatorikus ekvivalensét adjuk a 3.2 tételben és becsüljük ezt a „minimális számosságot".

3.2. DEFINÍCIÓ. Legyen $\mathscr{F} \subseteq P(\Omega) \times P(\Omega)$ teljes család. Jelölje ekkor $N(\mathscr{F})$ az $I(\mathscr{F})$ metszetirreducibilis elemeit; azaz

$$N(\mathscr{F}) = \{Y \in I(\mathscr{F}): Y \neq \cap\{Y' \in I(\mathscr{F}): Y' \supseteq Y, Y' \neq Y\}\}.$$

Azt mondjuk, hogy egy $M \subseteq I(\mathscr{F})$ *metszet-generálja* $I(\mathscr{F})$*-et,* ha

$$I(\mathscr{F}) = \{\cap M': M' \subseteq M\}.$$

3.1. LEMMA. Egy $M \subseteq I(\mathscr{F})$ akkor és csak akkor metszet-generálja $I(\mathscr{F})$-et, ha $N(\mathscr{F}) \subseteq M$.

Bizonyítás: A következő bizonyítás hálóelméletben jól ismert. Ha M metszet-generálja $I(\mathscr{F})$-et, akkor $N(\mathscr{F}) \subseteq M$ nyilvánvalóan. Azt kell még bizonyítanunk, hogy $N(\mathscr{F})$ metszetgenerálja $I(\mathscr{F})$-et. Tegyük fel, hogy ez nem igaz és legyen $X \in I(\mathscr{F})$ egy maximális számosságú, amelyre $X \neq \cap\{Y \in N(\mathscr{F}): Y \supseteq X\}$. Ekkor persze $X \notin N(\mathscr{F})$, azaz

(1) $X=\cap\{X' \in I(\mathscr{F}): X' \supseteq X, X' \neq X\}$.

Ha $X' \supseteq X$, $X' \neq X$, akkor $|X'|>|X|$ és így

(2) $X'=\cap\{Y \in N(\mathscr{F}): Y \supseteq X'\}$.

De az (1) és a (2) alapján $X=\cap\{Y \in N(\mathscr{F}): Y \supseteq X\}$, ami ellentmondás.

3.2. TÉTEL. Legyen $\mathcal{F}$ egy teljes család és $\mathcal{F}' \subseteq \mathcal{F}$. Ekkor $\mathcal{F}'$ minimális számosságú generátorrendszere $\mathcal{F}$-nek akkor és csak akkor, ha

$$(*) \qquad (\forall Y \in N(\mathcal{F}))(\exists A_Y \subseteq \Omega)(\mathcal{F}' = \{(A_Y, Y): Y \in N(\mathcal{F})\}).$$

Bizonyítás: Ha $\mathcal{F}'$ $(*)$ alakú, akkor nyilván generálja $\mathcal{F}$-et. Tegyük fel, hogy $\mathcal{F}'$ generálja $\mathcal{F}$-et. $(A, B) \in \mathcal{F}'$-re legyen B' maximális olyan, hogy $B' \supseteq B$ és $(A, B') \in \mathcal{F}$. Ekkor $\mathcal{F}'' = \{(A, B'): (A, B) \in \mathcal{F}'\}$ is generálja $\mathcal{F}$-et és $|\mathcal{F}''| \leqq |\mathcal{F}'|$. Nyilvánvaló, hogy $\{B': (\exists A)((A, B') \in \mathcal{F}'')\}$ metszetgenerálja $I(\mathcal{F})$-et, és így, 3.1 lemma szerint

$$|N(\mathcal{F})| \leqq |\mathcal{F}''| \leqq |\mathcal{F}'|.$$

továbbá, ha

$$|N(\mathcal{F})| = |\mathcal{F}''| = |\mathcal{F}'|,$$

akkor

$$N(\mathcal{F}) = \{B': (\exists A)((A, B') \in \mathcal{F}')\}.$$

3.3. KÖVETKEZMÉNY. Ha $\mathcal{F} \subseteq P(\Omega) \times P(\Omega)$ teljes család és $|\Omega| = n$, akkor létezik $\mathcal{F}'$ generátorrendszere $\mathcal{F}$-nek, melyre

$$|\mathcal{F}'| \leqq \frac{3}{2} \binom{n}{[n/2]}.$$

IRODALOM

[1] ARMSTRONG, W. W., "Dependency structures of data base relationships", *Information Processing 74*, North-Holland Publ. Co., Amsterdam, (1974) 580—583.

[2] ARMSTROKG, W. W., "On the generation of dependency structures of relational data bases", Publication #272, Université de Montréal, 1977.

[3] BÉKÉSSY, A. and DEMETROVICS, J., "Contribution to the theory of data base relations", *Discrete Mathematics* **27** (1979) 1—10.

[4] BÉKÉSSY, A., DEMETROVICS, J., HANNÁK, L., FRANKL, P. and KATONA, GY., "On the number of maximal dependencies in a data base relation of fixed order", *Discrete Mathematics* **30** (1980) 83—88.

[4] CODD, E. F., "A relational model of data for large shared data banks", *Comm. of ACM* **13** (1970) 377—387.

[6] CODD, E. F., "Further normalization of the data base relational model", *Courant Inst. Comp. Sci. Symp.* 6. Data Base Systems, Prentice-Hall, Englewood Cliffs, N. J. (1971) 33—64.

[7] DEMETROVICS, J., "Candidate keys and antichains", *SIAM J. on Algebraic and Discrete Methods* **1** (1980) 1—92.

[8] KLEITMAN, D., "On a combinatorial problem of Erdős, *Proc. AMS* (1966). 139—141.

[9] SPERNER, E., „Eine Satz über Untermengen einer endlichen Menge", *Math. Z.* **27** (1928) 544—548.

(Beérkezett: 1980. június 13.)

DEMETROVICS JÁNOS ÉS GYEPESI GYÖRGY
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1132 BUDAPEST, VICTOR HUGO U. 18.

GENERATION OF FULL FAMILIES AND THEIR REPREZENTATION BY RELATIONS

J. DEMETROVICS and GY. GYEPESI

In this paper we deal with two combinatorial problems of the theory of functional dependency. These two problems concern to the storage-problem of full families.

If R is a relation over the n-element set A then the cardinality of the functional dependencies fullfilled by R is between 3^n and 4^n.

Let $S(n)$ denote the smallest integer for which any full family $F \subseteq P(A) \times P(A)$ is the set of functional dependencies of a suitable relation over the n-element set A with at most $S(n)$ rows.

It is proved that $1/n^2 \binom{n}{[n/2]} \leq S(n) \leq 3/2 \binom{n}{[n/2]}$.

Hence it is economical to store a given full family by a suitable relation.

A variant of $S(n)$ is $s(n)$ where $s(n)$ is the smallest integer for which any *Sperner system* $C \subseteq P(A)$ is the set of candidate keys of a suitable relation over the n-element set A with at most $s(n)$ rows. We prove that $1/n^2 \binom{n}{[n/2]} \leq s(n) \leq \binom{n}{[n/2]} + 1$.

Finally we give a combinatorial characterization for the generating sets with minimal cardinality of full families. We prove that this "minimal cardinality" is less than $3/2 \binom{n}{[n/2]}$.

Alkalmazott Matematikai Lapok 6 (1980) 331–336

A PROGRAMOZÁS EGY REKURZÍV FÜGGVÉNYTANI MODELLJE I.

FÓTHI ÁKOS ÉS VARGA ZOLTÁN

Budapest

E dolgozatban a feladatorientált programozás egy matematikai modelljét kívánjuk megfogalmazni. Formalizáljuk a kifejthető feladat fogalmát és első lépésként igazoljuk, hogy minden kifejthető feladat megoldható egy, a gépi utasítások osztályára nézve relative rekurzív függvénynek megfelelő programmal.

1. Bevezetés

Az utóbbi években több progamozási modellt publikáltak, amelyekre a strukturált programozás alaptételeit igazolták (l. [1]-t és az ottani irodalomjegyzéket). A szerzők megmutatták, hogy miként lehet az adott modellben megfogalmazott feladatokat a strukturált programozás módszerével megoldani. Ezek a modellek általában olyanok, hogy a programot a számítógép működése alapján definiálják.

E. W. DIJKSTRA [2] könyve alapján felvetődik a feladat, hogy a programozás olyan matematikai modelljét konstruáljuk meg, amelyben a *Dijkstra-* és a *Jackson-féle* ([3]) *programozási eljárások* egyaránt természetes módon megfogalmazhatók. Ebben a dolgozatban egy ilyen modell alapjait fogalmazzuk meg. A feladat fogalmát formalizálva, egy feladatot kifejthetőnek nevezünk, ha véges sok megengedett kifejtési lépéssel olyan feladatokra vezethető vissza, amelyek már elemi programmal megoldhatók. A modell vizsgálatának első lépéseként *Church tézisére* való hivatkozás nélkül igazoljuk, hogy minden kifejthető feladathoz létezik olyan, az elemi programok osztályára nézve relatíve rekurzív függvény, amellyel, mint programmal az adott feladat megoldható.

Fogalmak és jelölések

Először bevezetünk néhány jelölést és fogalmat. Tetszőleges nem üres H és K halmaz esetén jelölje H^K a K halmazt a H halmazba képező függvények halmazát. Az $f: K \to H$ jelölés azt jelenti, hogy $f \in H^K$, $g: K \subset\!\to H$ pedig azt, hogy létezik olyan $\emptyset \neq M \subset K$ halmaz, hogy $g \in H^M$. Az f függvény értelmezési tartományát $D[f]$, értékkészletét pedig $R[f]$ jelöli. Tetszőleges $g: K \subset\!\to H$, $K_0 \subset D[g]$ és $H_0 \subset R[g]$ esetén jelölje $g(K_0)$ a K_0 halmaz képét, $g^{-1}(H_0)$ pedig H_0 ősképét.

Legyen most $\mathbf{N}$ a természetes számok halmaza, $\emptyset \neq N \subset M \subset \mathbf{N}$, és értelmezzük a következő függvényt:

$$\pi_N^M: H^M \to H^N, \quad \pi_N^M(x) := x|_N,$$

ahol $x|_N$ x-nek N-re való leszűkítése. Speciálisan legyen

$$\pi_M := \pi_M^{\mathbf{N}}, \quad \pi_k := \pi_{\{k\}} \quad (k \in \mathbf{N}).$$

A valamely függvényosztályra vonatkozóan rekurzív függvények értelmezéséhez legyen

$$\mathbf{N}_0 := \mathbf{N} \cup \{0\}, \quad \mathbf{M} := \{M \subset \mathbf{N},\ M \neq \emptyset,\ \text{véges}\}.$$

Legyen továbbá minden $k \in \mathbf{N}$ esetén

$$\sigma_k \colon \mathbf{N}_0^{\{k\}} \to \mathbf{N}_0^{\{k\}}, \quad \sigma_k(x) := x+1;$$

és minden $M \in \mathbf{M}$, $k \in \mathbf{N}$ esetén

$$\theta_{M,k} \colon \mathbf{N}_0^M \to \mathbf{N}_0^{\{k\}}, \quad \theta_{M,k}(x) = 0.$$

Legyen Σ olyan (a továbbiakban rögzített) függvényosztály, amelyre

(1.1) Minden $f \in \Sigma$ esetén létezik olyan $M, N \in \mathbf{M}$, hogy

$$f \colon \mathbf{N}_0^M \subset\!\!\to \mathbf{N}_0^N;$$

(1.2) Minden $M \in \mathbf{M}$, $k \in \mathbf{N}$ esetén $\theta_{M,k} \in \Sigma$;

(1.3) Minden $M, N \in \mathbf{M}$, $N \subset M$ esetén $\pi_N^M \in \Sigma$;

(1.4) Minden $k \in \mathbf{N}$ esetén $\sigma_k \in \Sigma$.

1.1. DEFINÍCIÓ: Valamely f függvény a Σ függvényosztályra vonatkozóan parciálisan rekurzívnak nevezünk, ha f véges számú lépésben előállítható Σ véges számú eleméből helyettesítéssel, primitív rekurzióval és minimalizálással. (Az utóbbi műveletekre nézve l. [4]-et.)

A Σ függvényosztályra vonatkozóan parciálisan rekurzív függvények halmazát a továbbiakban $R(\Sigma)$ jelöli.

1.2. DEFINÍCIÓ: Valamely H halmazt a Σ függvényosztályra vonatkozóan parciálisan rekurzívnak nevezünk, ha létezik olyan $M \in \mathbf{M}$ halmaz, hogy $H \subset \mathbf{N}_0^M$ és a H halmaz $\chi_H \colon \mathbf{N}_0^M \to \{0, 1\}$ karakterisztikus függvényére minden $k \in \mathbf{N}$ esetén $j_k \circ \chi_H \in R(\Sigma)$, ahol $j_k \colon \mathbf{N}_0 \to \mathbf{N}_0^{\{k\}}$ a természetes beágyazás.

A Σ függvényosztályra vonatkozóan parciálisan rekurzív halmazok összességét jelölje $RS(\Sigma)$.

2. A modell leírása

*Állapottér*nek nevezzük az $A := \mathbf{N}_0^{\mathbf{N}}$ halmazt, *állapotok*nak e halmaz elemeit. Tetszőleges $k \in \mathbf{N}$ esetén *k-adik változó*nak nevezzük a π_k projekciót. *Megengedett állapothalmaz*nak nevezünk egy $Q \subset A$ halmazt, ha van olyan $M \in \mathbf{M}$, hogy a $Q_M := \pi_M(Q)$ halmazra $Q_M \in RS(\Sigma)$ és $Q = \pi_M^{-1}(Q_M)$. Könnyen látható, hogy az utóbbi feltételeket kielégítő $M \in \mathbf{M}$ halmazok között egyértelműen létezik legszűkebb M_0. A továbbiakban legyen $\bar{Q} := Q_{M_0}$.

*Feladat*nak nevezünk egy (Q, R, ϱ) rendezett hármast, ha Q és R megengedett állapothalmaz, $\varrho \subset \bar{Q} \times \bar{R}$ és $D[\varrho] = \bar{Q}$. (Itt

$$D[\varrho] := \{x \in \bar{Q} \colon \exists y \in \bar{R},\ (x, y) \in \varrho\}.)$$

Valamely (Q, R, ϱ) *feladatot megoldó program*nak nevezünk egy $S \in R(\Sigma)$ függvényt, ha $\Gamma(S) \subset \varrho$, ahol $\Gamma[S]$ az S függvény grafikonja. Egy feladatot *megoldható*nak nevezünk, ha létezik e feladatot megoldó program.

Megjegyezzük, hogy az állapot fogalma a feladat, és nem a program állapotát tükrözi ([2]). Így az állapottér Σ-tól függetlenül, előre megadható. Általában a feladatot megoldó program nem függvény, hanem reláció, ebben a dolgozatban azonban csak a determinisztikus programokra szorítkozunk.

A program által igénybe vehető *„elemi programok"* halmazát egy olyan Σ függvényhalmazzal azonosítjuk, amely kielégíti az (1.1)—(1.4) feltételeket. Egy feladatot *elemi programmal megoldható*nak nevezünk, ha létezik a feladatot megoldó $S \in \Sigma$ program. A továbbiakban az elemi programmal megoldható feladatok halmazát $\mathscr{F}(\Sigma)$-val jelöljük.

A programozás során egy adott feladatot úgy kívánunk megoldani, hogy bizonyos megengedett kifejtési lépések sorozatával olyan feladatokra vezetjük vissza, amelyek már elemi programmal megoldhatók. A *megengedett kifejtési lépések* a következők: a dekompozíció, a szétválasztás és a ciklikus kifejtés. Ezeket a következőképpen értelmezzük:

(i) *Dekompozíció:* Azt mondjuk, hogy a (Q, R, ϱ) feladat *dekompozíció*val a (Q_1, R_1, ϱ_1) és a (Q_2, R_2, ϱ_2) feladat egymásutánjára vezethető vissza, ha

$$\bar{Q}_1 = \bar{Q}, \quad \bar{R}_1 \subset \bar{Q}_2, \quad \bar{R}_2 \subset \bar{R}, \quad \varrho_1 \circ \varrho_2 \subset \varrho. \tag{2.1}$$

A szétválasztás és a ciklikus kifejtés formalizálásához predikátum is szükséges. Tetszőleges

$$\varphi\colon A \to \{0, 1\}$$

függvényre legyen

$$T[\varphi] := \{a \in A,\ \varphi(a) = 1\},$$

$$F[\varphi] := A \setminus T[\varphi].$$

A $\varphi\colon A \to \{0, 1\}$ függvényt *megengedett predikátum*nak nevezzük, ha $T[\varphi] \in RS(\Sigma)$.

Legyen most (Q, R, ϱ) tetszőleges feladat, $Q' \subset A$ megengedett állapothalmaz, φ_1, φ_2 pedig megengedett predikátum. Tegyük fel, hogy

$$T[\varphi_1] \cap T[\varphi_2] = \emptyset, \tag{2.2}$$

$$Q' \cap (T[\varphi_1] \cup T[\varphi_2]) \supset Q. \tag{2.3}$$

(ii) A (2.2) és (2.3) feltétel mellett azt mondjuk, hogy a (Q, R, ϱ) feladat a Q', φ_1 és φ_2 által generált *szétválasztás*sal a (Q_1, R_1, ϱ_1) és a (Q_2, R_2, ϱ_2) feladatra vezethető vissza, ha $i = 1, 2$-re

$$\bar{Q}_i = Q' \cap T[\varphi_i],$$

$$\bar{R}_i \subset \bar{R},$$

$$\varrho_i \cap (\bar{Q} \times \bar{R}_i) \subset \varrho.$$

A ciklikus kifejtés definíciójához legyen

$$(Q, R, \varrho) \quad \text{és} \quad (Q_0, R_0, \varrho_0) \tag{2.4}$$

tetszőleges feladat, φ megengedett predikátum, legyen továbbá

$$D[t] \supset \overline{R_0 \cap T[\varphi]}, \quad t\colon D[t] \to \mathbf{N}_0, \; t(\overline{R_0 \cap T[\varphi]}) \subset \mathbf{N}; \tag{2.5}$$

$$\bar{Q} \subset \bar{R}_0, \; D[\varrho_0] = \bar{Q}_0 = \overline{R_0 \cap T[\varphi]}; \tag{2.6}$$

$$\varrho_0 = \{(x, y) \in \bar{Q}_0 \times \bar{R}_0 \colon t(x) \prec t(y)\}; \tag{2.7}$$

$$\tilde{\varrho}_0 \subset \varrho. \tag{2.8}$$

Itt $\tilde{\varrho}_0$ a ϱ_0 relációnak a φ predikátum által generált lezártja.

(iii) A (2.4)—(2.8) feltételek teljesülése esetén azt mondjuk, hogy a (Q, R, ϱ) feladat a φ, (Q_0, R_0, ϱ_0) és t által generált *ciklikus kifejtés*sel a (Q', R', ϱ') feladatra vezethető vissza, ahol

$$Q' := Q, \quad R' := R_0 \cup (Q \cap F[\varphi]),$$

$$\varrho' := \varrho_0 \cup \{(x, x) \in Q \times Q \colon x \in Q \cap F[\varphi]\}.$$

2.1. DEFINÍCIÓ: Egy feladatot *kifejthetőnek* nevezünk, ha véges számú megengedett kifejtési lépéssel visszavezethető véges számú $\mathscr{F}(\Sigma)$-beli feladatra.

3. A megoldható és a kifejthető feladatok kapcsolata

3.1. TÉTEL: Minden kifejthető feladat megoldható.

Bizonyítás: A kifejthetőség és a megoldhatóság definíciója értelmében elegendő azt megmutatni, hogy ha egy feladat a megengedett kifejtési lépések bármelyikével is visszavezethető megoldható feladat(ok)-ra, akkor a kiindulási feladat is megoldható.

a) Tegyük fel, hogy a (Q, R, ϱ) feladat dekompozícióval a (Q_1, R_1, ϱ_1) és a (Q_2, R_2, ϱ_2) feladat egymásutánjára vezethető vissza. Legyen $S_i \in R(\Sigma)$ a (Q_i, R_i, ϱ_i) feladatot megoldó program $(i = 1, 2)$. Ekkor az $S := S_2 \circ S_1$ függvény helyettesítéssel adódik az S_1 és az S_2 függvényből, így $S \in R(\Sigma)$. Mivel $\Gamma[S_i] \subset \varrho_i$ $(i = 1, 2)$, ezért (2.1) alapján

$$\Gamma[S] = \Gamma[S_2 \circ S_1] \subset \varrho_2 \circ \varrho_1 \subset \varrho,$$

tehát S a (Q, R, ϱ) feladatot megoldó program.

b) Tegyük fel, hogy a (Q, R, ϱ) feladat a Q' halmaz és a φ_1, φ_2 predikátumok által generált szétválasztással a (Q_1, R_1, ϱ_1) és a (Q_2, R_2, ϱ_2) feladatra vezethető vissza. Legyenek $S_i \in R(\Sigma)$ $(i = 1, 2)$ e feladatokat megoldó programok. Mivel $i = 1, 2$-re $T[\varphi_i] \in RS(\Sigma)$, ezért $Q \cap T[\varphi_i] \in RS(R)$ és a

$$\chi_{Q \cap T[\varphi_i]}\colon Q \to \{0, 1\}$$

karakterisztikus függvényre tetszőleges $k \in \mathbf{N}$ esetén $j_k \circ \chi_{Q \cap T[\varphi_i]} \in R(\Sigma)$. Így az

$$S := S_1 \chi_{Q \cap T[\varphi_1]} + S_2 \chi_{Q \cap T[\varphi_2]}$$

függvényre ugyancsak $S \in R(\Sigma)$.

Végül $\Gamma[S_i] \subset \varrho_i$, így a szétválasztás definíciója alapján

$$\Gamma(S) \subset (\varrho_1 \cap (\bar{Q} \times \bar{R}_1)) \cup (\varrho_2 \cap (\bar{Q} \times \bar{R}_2)) \subset \varrho.$$

Tehát S a (Q, R, ϱ) feladatot megoldó program.

c) Legyen végül (Q, R, ϱ) a (2.4)—(2.8) alatti jelölésekkel valamely φ, (Q_0, R_0, ϱ_0) és t által generált ciklikus kifejtéssel visszavezethető a (Q', R', ϱ') feladatra. Tegyük fel, hogy $s \in R(\Sigma)$ a (Q_0, R_0, ϱ_0) feladatot megoldó program. Jelölje $\tilde{s}$ az s függvénynek a φ predikátum által generált lezártját. Nyilván $\Gamma[s] \subset \varrho_0$ miatt $\tilde{s} \subset \tilde{\varrho}_0 \subset \varrho$, tehát $\tilde{s}$ a (Q, R, ϱ) feladatot megoldó program.

Megmutatjuk, hogy $\tilde{s} \in R(\Sigma)$. A lezárt értelmezése, valamint (2.6) és (2.7) alapján $D[\tilde{s}] = \bar{Q}_0$. Így minden $x \in \bar{Q}_0$ elemhez van olyan $y \in \mathbf{N}_0$, amelyre $s^y(x) \in F[\varphi]$. Értelmezzük az

$$f\colon\ \bar{Q} \times \mathbf{N}_0 \to \bar{R}_0, \quad f(x, y) := s^y(x)$$

függvényt. Ekkor $f \in R(\Sigma)$, mivel előáll a következő primitív rekurzióval:

$$f(x, 0) = x \quad (x \in \bar{Q}_0),$$

$$f(x, y+1) = s(f(x, y)) \quad (x \in Q,\ y \in \mathbf{N}_0).$$

Ezért az f-nek a $\varphi \in R(\Sigma)$ függvénybe való helyettesítésével adódó

$$P\colon\ \bar{Q}_0 \times \mathbf{N}_0 \to \mathbf{N}_0,$$

$$P(x, y) := \varphi(s^y(x))$$

függvényre minden $k \in \mathbf{N}$ esetén $j_k \circ P \in R(\Sigma)$. Így a P függvényből a minimalizálás műveletével adódó

$$p\colon\ \bar{Q}_0 \to \mathbf{N}_0,$$

$$p(x) := \min\{y \in \mathbf{N}_0\colon\ P(x, y) = 0\}$$

függvényre is minden $k \in \mathbf{N}$ esetén $j_k \circ p \in R(\Sigma)$. Végül, mivel nyilván

$$\tilde{s}(x) = s^{p(x)}(x) \quad (x \in \bar{Q}_0),$$

azért $\tilde{s}$ a p-nek f-be való helyettesítésével adódik. Ebből következik, hogy $\tilde{s} \in R(\Sigma)$.

Egy további dolgozatban igazoljuk, hogy az itt ismertetett modell elegendően komprehenzív abban az értelemben, hogy a kifejthető feladatok felölelik az összes „algoritmikusan megoldható" feladatot. Megmutatjuk továbbá, hogy e modellel miként valósítható meg a *Jackson-féle adatfinomításon alapuló programozási módszer* és a *Hoare-féle típuselmélet* összekapcsolása.

IRODALOM

[1] MILLS, H. D., Mathematical functions for structured programming", FSC 72—6012, IBM, Gaithersburg, Md., 1972.
[2] DIJKSTRA, E. W., *A Discipline of Programming* (Engl. Cliffs, 1976).
[3] JACKSOK, M. A., *Principles of Programming Design* (Academic Press, 1975).
[4] MALCEV, A. J., *Algorithmen und rekursive Funktionen* (Akademie-Verlag, Berlin, 1974).

(Beérkezett: 1980. november 27.)

FÓTHI ÁKOS ÉS VARGA ZOLTÁN
ELTE TTK NUMERIKUS ÉS GÉPI MATEMATIKA TANSZÉK
1088 BUDAPEST, MÚZEUM KRT. 6—8.

MODELLING PROGRAMS VIA RECURSIVE FUNCTIONS, I.

Á. FÓTHI and Z. VARGA

The presented model is aimed at a unified approach to the *programming methods of Dijkstra and Jackson.* The concept of an expansible problem is formalized. As a first step, it is proved that any expansible problem can be solved by a program defined as a function which is recursive with respect to the class of "terminal functions" (machine programs).

Alkalmazott Matematikai Lapok **6** (1980) 337—344

EGY HŐÁTADÁSI PROBLÉMA VIZSGÁLATA

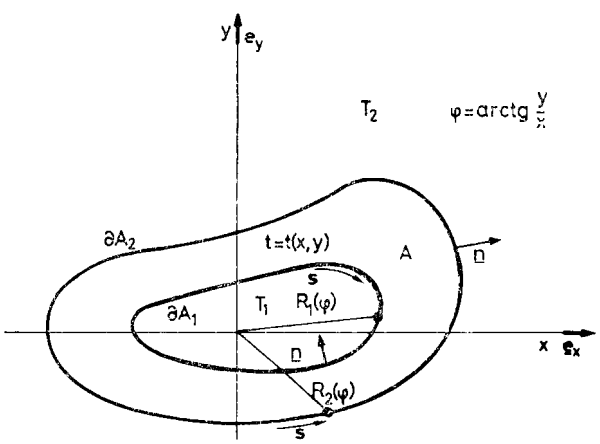
ECSEDI ISTVÁN

Miskolc

A műszaki gyakorlatban igen gyakran alkalmaznak magas hőmérsékletű folyadékok szállítására különböző keresztmetszetű csöveket, ezért nagy jelentősége van azoknak a formuláknak, amelyek lehetővé teszik a cső falán keresztül a magas hőmérsékletű folyadékból a külső környezetbe elvezetett hőmennyiség becslését. E tanulmány alsó és felső korlátokat ismertet az egységnyi hosszúságú csőszakaszon időegység alatt a külső környezetbe elvezetett hőmennyiség számértékére.

1. Bevezetés

Az 1. ábra egy végtelen hosszú üreges hengeres testet (csövet) szemléltet. A cső keresztmetszete a kétszeresen összefüggő xy síkbeli A tartomány, melynek belső határgörbéje a ∂A_1 zárt görbe, külső határgörbéje pedig a ∂A_2 zárt görbe. Az üreges test xy síkba eső keresztmetszetét és a keresztmetszethez kapcsolódó fontosabb mennyiségeket is az 1. ábra szemlélteti. A szilárd anyagú hengeres test T_1 hőmérsékletű folyadék elvezetését biztosítja. A T_1 hőmérsékletű folyadék és a hengeres test, valamint a T_2 hőmérsékletű külső közeg között a hőátadás konvenció útján valósul meg feltételezés szerint [1]. A szilárd test belsejében történő belső hővezetést a λ hővezetési együttható, a belső és külső csőfelületekkel kapcsolatos hőátadást pedig az α_1 és α_2 hőátadási tényezők jellemzik. Feltételezés szerint T_1, T_2 állandó, továbbá $T_{12} = T_1 - T_2 > 0$.



1. ábra.
Üreges alakú hengeres test

A fenti feltevések következménye, hogy a szilárd anyagú hengeres test hőmérséklet-eloszlása független a z koordinátától, vagyis a test $P(x, y, z)$ pontjában az állandósult állapothoz tartozó t hőmérséklet csak az x, y koordináták függvénye lesz, azaz $t=t(x, y)$.

A hővezetés *Fourier-féle elmélete* szerint a hengeres test állandósult állapothoz tartozó $t=t(x, y)$ hőmérséklet mezejét a következő peremérték feladattal tudjuk kapcsolatba hozni, ha a termikus paraméterek és sűrűség értékét állandónak tekintjük ([1], [2]):

$$\Delta t = 0 \quad (x, y)\in A, \tag{1.1}$$

$$\lambda\frac{\partial t}{\partial n}+\alpha_1(t-T_1) = 0 \quad (x, y)\in\partial A_1, \tag{1.2}$$

$$\lambda\frac{\partial t}{\partial n}+\alpha_2(t-T_2) = 0 \quad (x, y)\in\partial A_2. \tag{1.3}$$

A fenti egyenletekben:

x, y derékszögű koordináták,

$\Delta=\nabla\cdot\nabla=\dfrac{\partial^2}{\partial x^2}+\dfrac{\partial^2}{\partial y^2}$ *Laplace-féle differenciáloperátor,*

„$\cdot$" két vektor skaláris szorzatának jele, vagyis

$$\mathbf{a}\cdot\mathbf{b} = a_x b_x+a_y b_y, \quad \text{amennyiben} \quad \mathbf{a} = a_x\mathbf{e}_x+a_y\mathbf{e}_y, \quad \mathbf{b} = b_x\mathbf{e}_x+b_y\mathbf{e}_y,$$

$$\mathbf{a}^2 = \mathbf{a}\cdot\mathbf{a} = a_x^2+a_y^2,$$

$\nabla=\dfrac{\partial}{\partial x}\mathbf{e}_x+\dfrac{\partial}{\partial y}\mathbf{e}_y$ *Hamilton-féle differenciáloperátor,*

$\mathbf{e}_x$, $\mathbf{e}_y$ egységvektorok,

$\dfrac{\partial}{\partial n}$ a $\partial A=\partial A_1+\partial A_2$ határgörbe külső normálisa mentén számolt derivált jele.

Az egységnyi magasságú „belső" határfelületen időegység alatt

$$Q = \lambda\int_{\partial A_1}\frac{\partial t}{\partial n}\,ds \tag{1.4}$$

nagyságú hőmennyiség lép be az üreges hengeres testbe. Az egységnyi magasságú „külső" határoló felületszakaszon pedig időegység alatt

$$Q^* = \lambda\int_{\partial A_2}\frac{\partial t}{\partial n}\,ds \tag{1.5}$$

nagyságú hőmennyiség távozik a T_2 hőmérsékletű közegbe.

Könnyen belátható, hogy

$$Q+Q^* = 0. \tag{1.6}$$

Valóban az (1.1) egyenlet alapján írhatjuk, hogy

$$\int_A \Delta t\,dA = \int_{\partial A_1}\frac{\partial t}{\partial n}\,ds+\int_{\partial A_2}\frac{\partial t}{\partial n}\,ds. \tag{1.7}$$

Az (1.4), (1.5), (1.7) egyenletek kombinálásával közvetlenül a bizonyítandó (1.6) egyenletet nyerjük.

E tanulmány célja olyan egyenlőtlenségi relációk levezetése, melyek felhasználásával alsó és felső korlátokat tudunk képezni az (1.4) formula alapján meghatározható Q hőmennyiség (hőáram) számára. A bizonyított korlátok azért jelentősek, mert Q pontos értékének meghatározásához az (1.1), (1.2), (1.3) egyenletek által kijelölt peremérték feladatot kell megoldanunk az 1. ábrán vázolt kétszeresen összefüggő tartományra. Ez utóbbi probléma zárt alakú megoldása (beleértve a megoldás függvény végtelen soralakban való előállítását) néhány speciális esettől eltekintve nem is lehetséges.

Következőkben az (1.4) formula átalakításával foglalkozunk. A szorzat függvény deriválási szabályának és a *Gauss-féle integrálátalakítási tétel* együttes alkalmazásával nyerjük az (1.8) egyenletet:

$$\int_A (\nabla t)^2 \, dA = \int_{\partial A_1} t \frac{\partial t}{\partial n} \, ds + \int_{\partial A_2} t \frac{\partial t}{\partial n} \, ds. \tag{1.8}$$

A $\partial A = \partial A_1 + \partial A_2$ határgörbén mért ívkoordinátát s jelöli. Az (1.8) egyenlet és az (1.2), (1.3) peremfeltételek kombinálásával kapjuk az (1.9) egyenletet:

$$\int_A (\nabla t)^2 \, dA = T_1 \int_{\partial A_1} \frac{\partial t}{\partial n} \, ds - \frac{\lambda}{\alpha_1} \int_{\partial A_1} \left(\frac{\partial t}{\partial n}\right)^2 ds + \tag{1.9}$$

$$+ T_2 \int_{\partial A_2} \frac{\partial t}{\partial n} \, ds - \frac{\lambda}{\alpha_2} \int_{\partial A_2} \left(\frac{\partial t}{\partial n}\right)^2 ds.$$

Az (1.9) egyenletből elemi átalakításokkal és az (1.7), (1.4) egyenletek alkalmazásával az alábbi formulát tudjuk levezetni:

$$Q = \frac{\lambda}{T_1 - T_2} \left\{ \int_A (\nabla t)^2 \, dA + \frac{\lambda}{\alpha_1} \int_{\partial A_1} \left(\frac{\partial t}{\partial n}\right)^2 ds + \frac{\lambda}{\alpha_2} \int_{\partial A_2} \left(\frac{\partial t}{\partial n}\right)^2 ds \right\}. \tag{1.10}$$

Az (1.10) formulából kiolvasható, hogy $T_1 > T_2$ esetén Q mindig pozitív.
A levezetések során még alkalmazni fogjuk az alábbi formulát:

$$Q = \frac{\lambda}{T_1 - T_2} \left\{ \int_A (\nabla t)^2 \, dA + \frac{\alpha_1}{\lambda} \int_{\partial A_1} (t - T_1)^2 \, ds + \frac{\alpha_2}{\lambda} \int_{\partial A_2} (t - T_2)^2 \, ds \right\}. \tag{1.11}$$

Ez utóbbi formula az (1.10) formulából nyerhető az (1.2) és (1.3) egyenletek felhasználásával.

2. Felső korlát

TÉTEL. Bármely az $A + \partial A$ zárt tartományban folytonos, az A tartományban folytonos differenciálható $g = g(x, y)$ kétváltozós függvénnyel fennáll a

$$Q \leqq \frac{\lambda}{T_1 - T_2} \left\{ \int_A (\nabla g)^2 \, dA + \frac{\alpha_1}{\lambda} \int_{\partial A_1} (g - T_1)^2 \, ds + \frac{\alpha_2}{\lambda} \int_{\partial A_2} (g - T_2)^2 \, ds \right\}. \tag{2.1}$$

egyenlőtlenségi reláció.

Bizonyítás. Legyen

$$\varphi(x, y) = g(x, y) - t(x, y) \quad (x, y) \in A + \partial A. \tag{2.2}$$

Elemi számolással kapjuk az alábbi eredményt:

$$\int_A (\nabla g)^2\, dA + \frac{\alpha_1}{\lambda} \int_{\partial A_1} (g - T_1)^2\, ds + \frac{\alpha_2}{\lambda} \int_{\partial A_2} (g - T_2)^2\, ds = \int_A (\nabla t)^2\, dA + \tag{2.3}$$

$$+ \frac{\alpha_1}{\lambda} \int_{\partial A_1} (t - T_1)^2\, ds + \frac{\alpha_2}{\lambda} \int_{\partial A_2} (t - T_2)^2\, ds + \int_A (\nabla \varphi)^2\, dA + \frac{\alpha_1}{\lambda} \int_{\partial A_1} \varphi^2\, ds +$$

$$+ \frac{\alpha_2}{\lambda} \int_{\partial A_2} \varphi^2\, ds + 2 \int_A \nabla t \cdot \nabla \varphi\, dA + 2 \frac{\alpha_1}{\lambda} \int_{\partial A_1} \varphi (t - T_1)\, ds + 2 \frac{\alpha_2}{\lambda} \int_{\partial A_2} \varphi (t - T_2)\, ds.$$

A *Gauss-féle integrálátalakítási tételből* és az (1.1) egyenletből következik, hogy

$$\int_A \nabla t \cdot \nabla \varphi\, dA = \int_{\partial A_1} \varphi \frac{\partial t}{\partial n}\, ds + \int_{\partial A_2} \varphi \frac{\partial t}{\partial n}\, ds. \tag{2.4}$$

Az (1.2), (1.3), (2.4) egyenletek felhasználásával kapjuk a

$$\int_A \nabla t \cdot \nabla \varphi\, dA + \frac{\alpha_1}{\lambda} \int_{\partial A_1} \varphi (t - T_1)\, ds + \frac{\alpha_2}{\lambda} \int_{\partial A_2} \varphi (t - T_2)\, ds = \tag{2.5}$$

$$= \int_{\partial A_1} \varphi \left[\frac{\partial t}{\partial n} + \frac{\alpha_1}{\lambda} (t - T_1) \right] ds + \int_{\partial A_2} \varphi \left[\frac{\partial t}{\partial n} + \frac{\alpha_2}{\lambda} (t - T_2) \right] ds = 0$$

egyenletet.

Ez utóbbi eredmény, valamint az (1.11) és (2.3) egyenletek felhasználásával közvetlenül a bizonyítandó (2.1) egyenlőtlenségi relációt nyerjük.

3. A (2.1) felső korlát élesítése

A (2.1) egyenlőtlenségi relációt $\dot{g} = pg(x, y)$ függvényre alkalmazva, ahol p tetszőleges valós paraméter, kapjuk az alábbi eredményt:

$$Q \leqq \frac{\lambda}{T_1 - T_2} \left\{ p^2 \left(\int_A (\nabla g)^2\, dA + \frac{\alpha_1}{\lambda} \int_{\partial A_1} g^2\, ds + \frac{\alpha_2}{\lambda} \int_{\partial A_2} g^2\, ds \right) - \right. \tag{3.1}$$

$$\left. - 2p \left(T_1 \frac{\alpha_1}{\lambda} \int_{\partial A_1} g\, ds + T_2 \frac{\alpha_2}{\lambda} \int_{\partial A_2} g\, ds \right) \right\} + \frac{\alpha_1 T_1^2 l_1 + \alpha_2 T_2^2 l_2}{T_1 - T_2}.$$

A fenti formulában

$$l_1 = \int_{\partial A_1} ds, \quad l_2 = \int_{\partial A_2} ds.$$

Rögzített $g = g(x, y)$ függvény esetén a (3.1) egyenlőtlenség jobb oldala a p paraméter függvénye, a p alkalmas megválasztásával minimálissá tehető. Az elemi

számításokat mellőzve a szóban forgó minimum alapján — feltéve, hogy g nem azonosan zérus az $A+\partial A$ tartományban — az alábbi felső korlátot tudjuk levezetni Q számára:

$$(3.2)\quad Q \leqq \frac{\alpha_1 T_1^2 l_1+\alpha_2 T_2^2 l_2}{T_1-T_2} - \frac{1}{T_1-T_2}\,\frac{\left(\alpha_1 T_1 \int\limits_{\partial A_2} g\,ds+\alpha_2 T_2 \int\limits_{\partial A_2} g\,ds\right)^2}{\lambda \int\limits_A (\nabla g)^2 dA+\alpha_1 \int\limits_{\partial A_1} g^2\,ds+\alpha_2 \int\limits_{\partial A_2} g^2\,ds}.$$

A fenti formulából kiolvasható a következő egyszerű szerkezetű felső korlát is

$$(3.3)\quad Q < \frac{\alpha_1 T_1^2 l_1+\alpha_2 T_2^2 l_2}{T_1-T_2}.$$

A (3.3) egyenlőtlenségi relációból általában igen durva felső korlát képezhető Q-ra, hiszen ha $T_1-T_2\to 0$, akkor a probléma természetéből adódóan $Q\to 0$, noha a (3.3) jobb oldala a végtelenbe tart.

4. Alsó korlát

TÉTEL. Legyen $\mathbf{b}=\mathbf{b}(x, y)=b_x(x, y)\mathbf{e}_x+b_y(x, y)\mathbf{e}_y$ olyan a zérus vektorral nem azonosan egyenlő kétdimenziós vektormező, amely kielégíti a

$$(4.1)\quad \nabla\cdot\mathbf{b}=0 \quad (x, y)\in A$$

parciális differenciálegyenletet.
Fennáll a

$$(4.2)\quad Q \geqq \lambda(T_1-T_2)\,\frac{\left(\int\limits_{\partial A_1} \mathbf{b}\cdot\mathbf{n}\,ds\right)^2}{\int\limits_A \mathbf{b}^2\,dA+\frac{\lambda}{\alpha_1}\int\limits_{\partial A_1}(\mathbf{b}\cdot\mathbf{n})^2\,ds+\frac{\lambda}{\alpha_2}\int\limits_{\partial A_2}(\mathbf{b}\cdot\mathbf{n})^2\,ds}$$

egyenlőtlenségi reláció.

Bizonyítás. A Schwarz egyenlőtlenségi reláció alapján írhatjuk, hogy

$$(4.3)\quad [\mathbf{a}, \mathbf{a}][\mathbf{b}, \mathbf{b}] \geqq ([\mathbf{a}, \mathbf{b}])^2,$$

megjegyezvén, hogy az $\mathbf{a}=a_x(x, y)\mathbf{e}_x+a_y(x, y)\mathbf{e}_y$ és a $\mathbf{b}=b_x(x, y)\mathbf{e}_x+b_y(x, y)\mathbf{e}_y$ kétdimenziós vektorok $[\mathbf{a}, \mathbf{b}]$ skaláris szorzatát a (4.2) egyenlőtlenségi relációban az alábbi előírással értelmezzük:

$$(4.4)\quad [\mathbf{a}, \mathbf{b}] = \int\limits_A \mathbf{a}\cdot\mathbf{b}\,dA+\frac{\lambda}{\alpha_1}\int\limits_{\partial A_1}(\mathbf{n}\cdot\mathbf{a})(\mathbf{n}\cdot\mathbf{b})\,ds+\frac{\lambda}{\alpha_2}\int\limits_{\partial A_2}(\mathbf{n}\cdot\mathbf{a})(\mathbf{n}\cdot\mathbf{b})\,ds.$$

Legyen

$$(4.5)\quad \mathbf{a}=\nabla t$$

a $\mathbf{b}=\mathbf{b}(x, y)$ vektormező pedig tegyen eleget a (4.1) parciális differenciálegyenletnek. Elemi számolással azt találjuk, hogy

$$(4.6)\quad Q=\frac{\lambda}{T_1-T_2}[\nabla t, \nabla t].$$

A $[\nabla t, \mathbf{b}]$ skaláris szorzat az alábbi módon alakítható át:

$$(4.7)\qquad [\nabla t, \mathbf{b}] = \int_A \nabla\cdot(t\mathbf{b})\,dA - \int_A t(\nabla\cdot\mathbf{b})\,dA + \frac{\lambda}{\alpha_1}\int_{\partial A_1}(\mathbf{b}\cdot\mathbf{n})\frac{\partial t}{\partial n}\,ds +$$

$$+\frac{\lambda}{\alpha_2}\int_{\partial A_2}(\mathbf{b}\cdot\mathbf{n})\frac{\partial t}{\partial n}\,ds = \int_{\partial A_1} t(\mathbf{b}\cdot\mathbf{n})\,ds + \int_{\partial A_2} t(\mathbf{b}\cdot\mathbf{n})\,ds + \frac{\lambda}{\alpha_1}\int_{\partial A_1}(\mathbf{b}\cdot\mathbf{n})\frac{\partial t}{\partial n}\,ds +$$

$$+\frac{\lambda}{\alpha_2}\int_{\partial A_2}(\mathbf{b}\cdot\mathbf{n})\frac{\partial t}{\partial n}\,ds = (T_1 - T_2)\int_{\partial A_1}\mathbf{b}\cdot\mathbf{n}\,ds.$$

A (4.7) egyenlet levezetésénél felhasználtuk, hogy

$$(4.8)\qquad \int_A \nabla\cdot\mathbf{b}\,dA = \int_{\partial A_1}\mathbf{b}\cdot\mathbf{n}\,ds + \int_{\partial A_2}\mathbf{b}\cdot\mathbf{n}\,ds = 0.$$

A (4.3), (4.5), (4.6), (4.6) formulák kombinálásával a bizonyítandó (4.1) egyenlőtlenségi relációt nyerjük.

5. Néhány korlát

Legyen a (3.2) egyenlőtlenségi relációban $g = c$ =állandó $(c \neq 0)$. Elemi számolással a következő eredményt nyerjük:

$$(5.1)\qquad Q \leqq \frac{T_1 - T_2}{\dfrac{1}{\alpha_1 l_1} + \dfrac{1}{\alpha_2 l_2}}.$$

Legyen a (3.2) egyenlőtlenségi relációban $g = g(x, y) = r = \sqrt{x^2 + y^2}$. A fenti függvénnyel számolva nyerjük az (5.2) felső korlátot Q számára:

$$(5.2)\qquad Q \leqq \frac{\alpha_1 T_1^2 l_1 + \alpha_2 T_2^2 l_2}{T_1 - T_2} -$$

$$-\frac{1}{T_1 - T_2}\,\frac{\left(\alpha_1 T_1 \int_{\partial A_1} r\,ds + \alpha_2 T_2 \int_{\partial A_2} r\,ds\right)^2}{\lambda f + \alpha_1 \int_{\partial A_1} r^2\,ds + \alpha_2 \int_{\partial A_2} r^2\,ds}, \qquad \left(f = \int_A dx\,dy\right)$$

Legyen a (4.2) egyenlőtlenségi relációban

$$(5.3)\qquad \mathbf{b} = \mathbf{b}(x, y) = \frac{x}{x^2 + y^2}\mathbf{e}_x + \frac{y}{x^2 + y^2}\mathbf{e}_y.$$

Könnyen verifikálható, hogy a

$$(5.4)\qquad \nabla\cdot\mathbf{b} = 0 \quad (x, y) \in A$$

egyenletet a fenti alakú $\mathbf{b} = \mathbf{b}(x, y)$ vektor kielégíti. Elemi, de kissé hosszadalmas számítással azt kapjuk, hogy

$$(5.5)\qquad \int_{\partial A_1}\mathbf{b}\cdot\mathbf{n}\,ds = 2\pi.$$

Legyen a ∂A_1 határgörbe egyenlete az $r\varphi$ polárkoordináta-rendszerben $r_1 = r_1(\varphi)$, a ∂A_2 határgörbe ugyanazon $r\varphi$ polárkoordináta-rendszerbeli egyenlete pedig $r_2 = r_2(\varphi)$ (1. ábra). Egyszerű számítással nyerjük a következő eredményeket:

$$\int_A \mathbf{b}^2 \, dA = \int_0^{2\pi} \ln \left[\frac{r_2(\varphi)}{r_1(\varphi)}\right] d\varphi, \tag{5.6}$$

$$\int_{\partial A_1} (\mathbf{b} \cdot \mathbf{n})^2 \, ds = \int_0^{2\pi} \frac{d\varphi}{\sqrt{[r_1(\varphi)]^2 + [r_1'(\varphi)]^2}}, \tag{5.7}$$

$$\int_{\partial A_2} (\mathbf{b} \cdot \mathbf{n})^2 \, ds = \int_0^{2\pi} \frac{d\varphi}{\sqrt{[r_2(\varphi)]^2 + [r_1'(\varphi)]^2}}. \tag{5.8}$$

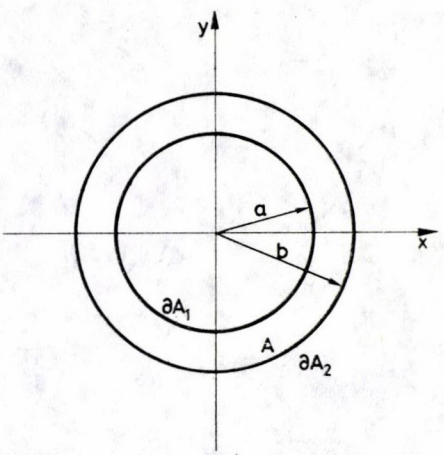
A fenti eredmények (4.2) formulába való helyettesítésével kapjuk az alábbi alsó korlátot a Q hőáram számára:

$$Q \geqq \frac{4\pi^2 (T_1 - T_2)\lambda}{\int_0^{2\pi} \left(\ln \frac{r_2}{r_1}\right) d\varphi + \frac{\lambda}{\alpha_1} \int_0^{2\pi} \frac{d\varphi}{\sqrt{r_1^2 + (r_1')^2}} + \frac{\lambda}{\alpha_2} \int_0^{2\pi} \frac{d\varphi}{\sqrt{r_2^2 + (r_2')^2}}}. \tag{5.9}$$

Abban a speciális, de a gyakorlat szempontjából fontos esetben, mikor $r_2 = \nu r_1$, ν = állandó, $\nu > 1$, az (5.9) formula a következő alakra hozható:

$$Q \geqq \frac{4\pi^2 (T_1 - T_2)\lambda}{2\pi \ln \nu + \left(\frac{\lambda}{\alpha_1} + \frac{1}{\nu} \frac{\lambda}{\alpha_2}\right) \int_0^{2\pi} \frac{d\varphi}{\sqrt{r_1^2 + (r_1')^2}}}. \tag{5.10}$$

A 2. ábra egy körgyűrű keresztmetszetű csövet szemléltet. Az (5.1) és (5.10) formulák



2. ábra.
Körgyűrű keresztmetszet

alkalmazásával kapjuk az alábbi eredményt:

$$2\pi \frac{T_1 - T_2}{\frac{1}{\lambda} \ln \frac{b}{a} + \frac{1}{\alpha_1 a} + \frac{1}{\alpha_2 b}} \leqq Q \leqq 2\pi \frac{T_1 - T_2}{\frac{1}{\alpha_1 a} + \frac{1}{\alpha_2 b}}. \tag{5.11}$$

A fenti formulából kiolvasható, hogy az alsó és felső korlát értéke $b \to a$ esetén egybeesik, vagyis

$$\tilde{Q} = 2\pi \frac{T_1 - T_2}{\frac{1}{\alpha_1 a} + \frac{1}{\alpha_2 b}} \tag{5.12}$$

formula alapján kiszámolt mennyiséghez alulról tart a Q hőáram, ha $b \to a$.

IRODALOM

[1] CARSLAW, H. S. and JEAGER, J. C., *Conduction of Heat in Solids* (University Press, Oxford, 1959).
[2] OZISIK, M. N., *Boundary Value Problems of Heat Conduction* (Scranton Pa. International Textbook, 1968).

(Beérkezett: 1980. november 6.)

ECSEDI ISTVÁN
NME MECHANIKA TANSZÉK
3515 MISKOLC, EGYETEMVÁROS

THE INVESTIGATION OF A PROBLEM OF HEAT TRANSFER

I. ECSEDI

The aim of the present paper is to give upper and lower bounds for the rate of flow of heat.
The formulas (2.1), (3.1), (3.3) yield the upper bound. The formula (4.2) yields the lower bound. Exemples illustrate the applications of the bounding formulas.

Alkalmazott Matematikai Lapok 6 (1980) 345—349

NEM SZIMMETRIKUS VÉLETLEN (0, 1) MÁTRIX SPEKTRUMÁNAK ASZIMPTOTIKUS VISELKEDÉSÉRŐL

JUHÁSZ FERENC

Budapest

Az [2] dolgozatban véletlen gráf (szimmetrikus mátrix) spektrumával foglalkoztunk. Megmutatjuk, hogy a fenti munka eredményei igazak nem szimmetrikus véletlen (0, 1) mátrixra (irányított gráfra) is, nevezetesen: a legnagyobb sajátérték n rendű, míg a többi sajátérték tetszőleges $\varepsilon>0$ esetén $o\left(n^{\frac{1}{2}+\varepsilon}\right)$ rendű mértékben. Az itt alkalmazott módszer elintézi a szimmetrikus esetet is.

1. TÉTEL. Legyen az $\mathbf{A}_n=(a_{ij})$ olyan $n\times n$ méretű mátrix, amelynek az elemei $i\neq j$ esetén teljesen független valószínűségi változók, $P(a_{ij}=1)=p$, $P(a_{ij}=0)=q=1-p$. Tegyük fel, hogy a főátlóban $a_{ii}\equiv 0$. Ha $\lambda=\lambda(n)$ az $\mathbf{A}_n$ legnagyobb sajátértéke, akkor tetszőleges $\varepsilon>0$ esetén

$$\frac{\lambda}{n}-p=o\left(n^{-\frac{1}{2}+\varepsilon}\right) \quad \text{mértékben.}$$

Bizonyítás. A *Perron—Frobenius-tétel* szerint

$$\min_{1\leqq i\leqq n}\sum_{j=1}^{n}a_{ij}\leqq\lambda\leqq\max_{1\leqq i\leqq n}\sum_{j=1}^{n}a_{ij}.$$

Ha rögzített i indexre alkalmazzuk a *Csebisev-egyenlőtlenség* éles alakját kapjuk, hogy

$$P\left(\left|\frac{1}{n}\sum_{j=1}^{n}a_{ij}-p\right|>Kn^{-\frac{1}{2}+\varepsilon}\right)\leqq 2\exp\left(-Cn^{2\varepsilon}\right).$$

Innen

$$P\left(\max_{1\leqq i\leqq n}\left|\frac{1}{n}\sum_{j=1}^{n}a_{ij}-p\right|>Kn^{-\frac{1}{2}+\varepsilon}\right)\leqq 2n\exp\left(-Cn^{2\varepsilon}\right)=o(1).$$

DEFINÍCIÓ. Az $f(n)$ és $g(n)$ valószínűségi változó sorozatról azt mondjuk, hogy $f(n)\leqq g(n)$ mértékben, ha minden $\delta>0$ számhoz van olyan n_0, hogy $n>n_0$ esetén

$$P\left(f(n)>g(n)\right)<\delta.$$

2. TÉTEL. Legyen az $\mathbf{A}_n$ mátrix olyan, mint az 1. tételben. Ha $\mathbf{u}=\mathbf{u}(n)$ a $\lambda=\lambda(n)$ legnagyobb sajátértékhez tartozó sajátvektor, oly módon, hogy $\max_{1\leqq i\leqq n}u_i=1$, akkor $\varepsilon>0$ esetén $\max_{1\leqq i\leqq n}(1-u_i)=o\left(n^{-\frac{1}{2}+\varepsilon}\right)$ mértékben.

Bizonyítás. Legyen $\min\limits_{1\leqq i\leqq n} u_i=u_{\min}$, $\max\limits_{1\leqq i\leqq n} u_i=u_{\max}=1$.
Vezessük be a következő jelöléseket:

$$m = m(\varepsilon) = \sum_{u_i<1-n^{-\frac{1}{2}+\varepsilon}} 1,$$

$$d_i = \frac{1}{n}\sum_{j=1}^{n} a_{ij}, \quad d_{i1} = \frac{1}{m}\sum_{u_j<1-n^{-\frac{1}{2}+\varepsilon}} a_{ij}, \quad d_{i2} = \frac{1}{n-m}\sum_{u_j\geqq 1-n^{-\frac{1}{2}+\varepsilon}} a_{ij}.$$

Legyen $0<k_1<p$, $q<k_2=1-k_1$. Az $\mathbf{A}_n$ véletlen mátrix tetszőleges megvalósulására a következő három eset közül pontosan egy teljesül:

(a) $m>k_2 n$,
(b) $k_1 n\leqq m\leqq k_2 n$,
(c) $m<k_1 n$.

Az (a) esetben

$$\lambda = \sum_{j=1}^{n} a_{\max j}u_j \leqq m d_{\max 1}\left(1-n^{-\frac{1}{2}+\varepsilon}\right)+(n-m)d_{\max 2}.$$

Innen

$$d_{\max 1} \leqq \frac{n\left(d_{\max}-\frac{\lambda}{n}\right)}{mn^{-\frac{1}{2}+\varepsilon}} \leqq \frac{d_{\max}-\frac{\lambda}{n}}{k_2 n^{-\frac{1}{2}+\varepsilon}}.$$

A számláló az 1. tétel következtében $o\left(n^{-\frac{1}{2}+\frac{\varepsilon}{2}}\right)$ rendű mértékben, amiért is $d_{\max 1}=$ $=o\left(n^{-\frac{\varepsilon}{2}}\right)$ mértékben. Ennek a valószínűsége azonban az $m>k_2 n>qn$ feltétel miatt tart nullához.

A (b) esetben legyen i olyan index, amelyre $u_i\geqq 1-n^{-\frac{1}{2}+\frac{\varepsilon}{2}}$.
Ekkor

$$\lambda u_i = \sum_{j=1}^{n} a_{ij}u_j \leqq m d_{i1}\left(1-n^{-\frac{1}{2}+\varepsilon}\right)+(n-m)d_{i2}.$$

Innen

$$d_{i1} \leqq \frac{nd_i-\lambda u_i}{mn^{-\frac{1}{2}+\varepsilon}} \leqq \frac{\left(d_i-\frac{\lambda}{n}\right)+\frac{\lambda}{n}(1-u_i)}{k_1 n^{-\frac{1}{2}+\varepsilon}}.$$

A számláló első tagja az 1. tétel, a második tagja az i index fenti választása miatt $o\left(n^{-\frac{1}{2}+\frac{\varepsilon}{2}}\right)$ rendű mértékben. Ezért $d_{i1}=o\left(n^{-\frac{\varepsilon}{2}}\right)$ mértékben. Ha a szóban forgó i indexek száma $\sum\limits_{u_i\geqq 1-n^{-\frac{1}{2}+\frac{\varepsilon}{2}}} 1=n-m\left(\frac{\varepsilon}{2}\right)\geqq k_1 n$, akkor a mátrixnak egy $k_1 n\times k_1 n$

méretű részében az egyesek sűrűsége tart nullához. Ennek a valószínűsége azonban nullához tart. Amennyiben $n-m\left(\frac{\varepsilon}{2}\right)<k_1 n$ azaz $m>k_2 n$, úgy esetünket visszavezettük az (a) esetre.

Az esetek nagy többsége tehát a (c) ponthoz tartozik.

Ekkor

$$\lambda u_{\min} = \sum_{j=1}^{n} a_{\min j} u_j \geqq m d_{\min 1} u_{\min} + (n-m) d_{\min 2}\left(1 - n^{-\frac{1}{2}+\varepsilon}\right).$$

Innen

$$u_{\min} \geqq \frac{(n-m) d_{\min 2}\left(1 - n^{-\frac{1}{2}+\varepsilon}\right)}{(\lambda - n d_{\min}) + (n-m) d_{\min 2}}.$$

Minthogy $m<k_1 n$ ezért $n-m>k_2 n>qn$. Ennélfogva $d_{\min 2}>\alpha$ mértékben, ahol $\alpha>0$ rögzített szám. Ezért a fenti nevező nem tűnik el.

$$u_{\min} \geqq \frac{1 - n^{-\frac{1}{2}+\varepsilon}}{1 + \frac{\frac{\lambda}{n} - d_{\min}}{k_2 \alpha}} \geqq 1 - n^{-\frac{1}{2}+2\varepsilon} \quad \text{mértékben.}$$

3. TÉTEL. Legyen $\mathbf{A}_n$ olyan, mint az 1. tételben. Ekkor az $\mathbf{A}_n$ mátrix legnagyobbtól különböző sajátértékei $\varepsilon>0$ esetén $o\left(n^{\frac{1}{2}+\varepsilon}\right)$ rendűek mértékben.

Bizonyítás. Legyen $\lambda=\lambda(n)$ az $\mathbf{A}_n$ mátrix legnagyobb sajátértéke, $\mathbf{v}=(v_i)$, illetve $\mathbf{u}=(u_i)$ $(v_{\max}=u_{\max}=1)$ a hozzá tartozó sajátvektor balról, illetve jobbról. Ekkor a $\mathbf{B}_n=\mathbf{A}_n-\frac{\lambda}{(\mathbf{u},\mathbf{v})}\mathbf{u}\mathbf{v}^T$ mátrix sajátértékei az $\mathbf{A}_n$ mátrix legnagyobbtól különböző sajátértékei, továbbá a nulla.

Bontsuk fel a $\mathbf{B}_n$ mátrixot a következő módon:

$$\mathbf{B}_n = \mathbf{C}_n + \mathbf{D}_n + \mathbf{E}_n + \mathbf{F}_n,$$

ahol

$$\mathbf{C}_n = (c_{ij}), \quad c_{ij} = a_{ij} - p,$$

$$\mathbf{D}_n = (d_{ij}), \quad d_{ij} = p - \frac{\lambda}{n},$$

$$\mathbf{E}_n = (e_{ij}), \quad e_{ij} = \frac{\lambda}{n}\left(1 - \frac{n}{(\mathbf{u},\mathbf{v})}\right),$$

$$\mathbf{F}_n = (f_{ij}), \quad f_{ij} = \frac{\lambda}{(\mathbf{u},\mathbf{v})}(1 - u_i v_j).$$

Jelölje $\| \ . \ \|$ a mátrix euklideszi normáját. Ekkor

$$n^{-\frac{1}{2}-\varepsilon}\|\mathbf{D}_n\| = n^{-\frac{1}{2}-\varepsilon} n\left|p-\frac{\lambda}{n}\right| = n^{\frac{1}{2}-\varepsilon} o\left(n^{-\frac{1}{2}+\frac{\varepsilon}{2}}\right) = o\left(n^{-\frac{\varepsilon}{2}}\right) \quad \text{mértékben,}$$

$$n^{-\frac{1}{2}-\varepsilon}\|\mathbf{E}_n\| \leqq n^{-\frac{1}{2}-\varepsilon} n \frac{n-(\mathbf{u},\mathbf{v})}{(\mathbf{u},\mathbf{v})} \leqq n^{\frac{1}{2}-\varepsilon} \frac{n-n\left(1-o\left(n^{-\frac{1}{2}+\frac{\varepsilon}{2}}\right)\right)^2}{n\left(1-o\left(n^{-\frac{1}{2}+\frac{\varepsilon}{2}}\right)\right)^2} \leqq$$

$$\leqq n^{\frac{1}{2}-\varepsilon} \frac{2o\left(n^{-\frac{1}{2}+\frac{\varepsilon}{2}}\right)}{\frac{1}{4}} = o\left(n^{-\frac{\varepsilon}{2}}\right) \quad \text{mértékben,}$$

$$n^{-\frac{1}{2}-\varepsilon}\|\mathbf{F}_n\| \leqq n^{-\frac{1}{2}-\varepsilon}\sqrt{n^2\left(1-\left(1-o\left(n^{-\frac{1}{2}+\frac{\varepsilon}{2}}\right)\right)^2\right)^2} \leqq$$

$$\leqq n^{\frac{1}{2}-\varepsilon} 2o\left(n^{-\frac{1}{2}+\frac{\varepsilon}{2}}\right) = o\left(n^{-\frac{\varepsilon}{2}}\right) \quad \text{mértékben.}$$

A $\|\mathbf{C}_n\|$ becslésére álljon itt a következő

4. TÉTEL. Legyen az $\mathbf{A}_n=(a_{ij})$ olyan $n\times n$ méretű mátrix, amelynek elemei $i\neq j$ esetén teljesen független azonos elosztású valószínűségi változók. Tegyük fel, hogy $Ea_{ij}=0$, továbbá, hogy a_{ij} összes momentuma létezik. A főátlóban legyen $a_{ii}\equiv 0$. Ekkor ha $\lambda=\lambda(n)$ az $\mathbf{A}_n$ mátrix maximális sajátértéke, akkor tetszőleges $\varepsilon>0$ esetén $\lambda=o\left(n^{\frac{1}{2}+\varepsilon}\right)$ mértékben.

Bizonyítás. A gondolatmenet az [1], [2], [3] dolgozatokban foglaltakon alapszik. Legyen $\mathbf{S}=\mathbf{A}_n\mathbf{A}_n^T$, ahol $\mathbf{A}^T$ a transzponált mátrixot jelöli. Ha $\mu_1\geqq\mu_2\geqq\ldots\geqq\mu_n$ az $\mathbf{S}$ mátrix sajátértékei, akkor $|\lambda|\leqq\sqrt{\mu_1}$. Ekkor

$$M_{k,n} = n^{-1-k}\sum_{i=1}^{n}\mu_i^k =$$

$$= n^{-1-k}\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\ldots\sum_{i_k=1}^{n}\sum_{j_1=1}^{n}\sum_{j_2=1}^{n}\ldots\sum_{j_k=1}^{n}(a_{i_1j_1}a_{i_2j_1})(a_{i_2j_2}a_{i_3j_2})\ldots(a_{i_kj_k}a_{i_1j_k}).$$

Bontsuk fel a fenti összeget részösszegekre $M_{k,n}=n^{-1-k}\sum_{i=1}^{n}N_i$, ahol N_i azon szorzatokat tartalmazza, melyekben i különböző index szerepel. Vizsgáljuk meg rögzített, i számú indexhez tartozó szorzatok várható értékét. Minthogy $\binom{n}{i}$-féle ilyen indexhalmaz létezik, ezért ha $i\leqq k$, akkor $\lim_{n\to\infty} n^{-1-k}EN_i=0$. Másrészt ahhoz, hogy egy szorzat várható értéke ne tűnjön el, minden tényezőnek legalább kétszer kell szerepel-

nie, azaz legfeljebb $(k+1)$ különböző index fordulhat elő benne. Ily módon kapjuk, hogy $E\sum_{i=1}^{n}\mu_i^k = O(n^{k+1})$, azaz $E\mu_1^k = O(n^{k+1})$.

Innen

$$P\left(|\lambda| > n^{\frac{1}{2}+\varepsilon}\right) < P(\mu_1^k > n^{k+2k\varepsilon}) = O(n^{-(2k\varepsilon-1)}) = o(1),$$

ha k elegendően nagy.

IRODALOM

[1] ARNOLD, L., "On the asymptotic distribution of the eigenvalues of random matrices", *J. Math. Analysis Appl.* **20** (1967) 262—268.

[2] JUHÁSZ, F., "On the spectrum of a random graph" *Algebraic Methods in Graph Theory* eds. L. Lovász, Vera T. Sós, Colloquia Mathematica Societatis János Bolyai 25 (North-Holland, Amsterdam, 1981) 313—317.

[3] WIGNER, E. P., "Characteristic vectors of bordered matrices with infinite dimensions", *Annals of Mathematics* **62** (1955) 548—564.

(Beérkezett: 1980. szeptember 30.)

JUHÁSZ FERENC
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST, ÚRI U. 49.

ON THE ASYMPTOTIC BEHAVIOUR OF THE SPECTRA OF NON-SYMMETRIC RANDOM (0, 1) MATRICES

F. JUHÁSZ

In [2] we investigated the spectrum of a random graph (symmetric matrix). In the present paper we are going to show that these results carry over to non-symmetric random (0, 1) matrices (directed graphs), namely the largest eigenvalue is of order n, while the other eigenvalues are of order $n^{\frac{1}{2}+\varepsilon}$ ($\varepsilon>0$ arbitrary). The method used applies to the symmetric case, too.

Alkalmazott Matematikai Lapok 6 (1980) 351—392

FORGALOMELOSZTÁS MEGOLDÁSA SZÁMÍTÓGÉPPEL

BAKÓ ANDRÁS

Győr

Közúti és városi úthálózatok távlati tervezésének és a jelenlegi forgalmi helyzet analizálásának egyik legfontosabb lépése a forgalom hálózatra történő elosztása.

Két alaphipotézis alapján fogalmazhatunk meg elosztási modelleket, amelyek optimalizációs feladatokhoz vezetnek. Megoldásuk rendszerint — az úthálózat méretei miatt — heurisztikus módszerrel történik. A pontos és a heurisztikus eljárások egyaránt felhasználják a legrövidebb út feladatot. A dolgozatban összefoglaljuk a fontosabb forgalomszétosztási és legrövidebb út eljárásokat, és kitérünk a számítástechnikai tapasztalatokra is.

1. Forgalomelosztási feladat

1.1. *Bevezetés, a feladat megfogalmazása*

Közúti és városi úthálózatok távlati tervezésének és a jelenlegi forgalmi helyzet analizálásának egyik legfontosabb lépése a forgalom hálózatra történő elosztása.

Két alaphipotézis alapján fogalmazhatunk meg elosztási modelleket, amelyek optimalizációs feladatokhoz vezetnek. Megoldásuk rendszerint — az úthálózat méretei miatt — heurisztikus módszerrel történik. A pontos és a heurisztikus eljárások egyaránt felhasználják a legrövidebb út feladatot. A dolgozatban összefoglaljuk a fontosabb forgalomszétosztási és legrövidebb út eljárásokat, és kitérünk a számítástechnikai tapasztalatokra is.

Az eljárások számítógépes megvalósításával hazánkban az *MTA Számítástechnikai Központjának Operációkutatási Osztálya* és annak jogutódja a *SZTAKI* foglalkozott először. A közlekedési modelleket a pécsi *ÉGSZI* és az *Útügyi Kutató Intézet* és a *BME Útépítési Tanszékének* munkatársai alkalmazták kezdetben. Később a *KÖTUKI*, az *UVATERV*, a *VATI*, a *METROBER* és a *KTMF* is bekapcsolódott a munkába. A *KPM* hamar felismerte az alkalmazási lehetőséget és számos kutatási munkát finanszírozott és támogat jelenleg is.

Az úthálózat egy irányított gráfnak felel meg. Jelölje az úthálózat közlekedési csomópontjainak véges halmazát $N=\{1, 2, \ldots, n\}$, az irányított útszakaszainak halmazát $E=\{(i,j)\}$. Az úthálózatot tervezési szempontból körzetekre bontjuk. Minden körzethez hozzárendelünk egy forrás- vagy nyelőpontot, esetleg mindkettőt. Jelölje a források halmazát S, a nyelőkét T, és legyen $p=\|S\|$, $q=\|T\|$, $p\leqq n$, $q\leqq n$. Forgalomszámlálásból vagy a forgalom előrebecsléséből kapjuk a $\mathbf{H}=(h_{ij})$ forgalmi mátrixot, melynek h_{st} eleme az s forráspontból a t nyelőpontba irányuló forgalmat mutatja. Az (i,j) élen menő x_{ij} forgalmat (folyamot) nem ismerjük. Jelölje x_{ij}^{st} az $s\in S$ forrásból a $t\in T$ nyelőbe menő forgalmat az (i,j) élen.

Az x_{ij} folyam az (i, j) élen az összes forrásból összes nyelőbe menő forgalomból tevődik össze, azaz:

$$x_{ij} = \sum_{\substack{s \in S \\ t \in T}} x_{ij}^{st}$$

Legyen $P_{st}=(s=i_0, i_1, \ldots, i_r=t)=((i_0, i_1), (i_1, i_2), \ldots (i_{r-1}, i_r))$ az s pontból a t pontba vezető út. Rendeljünk a hálózat minden $(i, j) \in E$ éléhez egy t_{ij} élhosszat és egy $k_{ij}(x_{ij})$ egységnyi szállítási költséget és egy b_{ij} kapacitáskorlátot.

A P_{st} út $l(P_{st})$ hosszán és a $k(P_{st})$ egységköltségén az alábbiakat értjük:

$$l(P_{st}) = \sum_{(i,j) \in P_{st}} t_{ij}, \quad k(P_{st}) = \sum_{(i,j) \in P_{st}} k_{ij}(x_{ij}). \tag{1.1}$$

Az $x=(x_{ij})$ folyamot lehetséges folyamnak nevezzük, ha

$$0 \leqq x_{ij} \leqq b_{ij} \tag{1.2}$$

és

$$\sum_{(i,j) \in E} x_{ij}^{st} - \sum_{(i,j) \in E} x_{ji}^{st} = \begin{cases} h_{st}, & \text{ha} \quad i = s \\ -h_{st}, & \text{ha} \quad i = t \\ 0, & \text{egyébként.} \end{cases} \tag{1.3}$$

Az x folyam költségén, vagy másképpen a rendszer összköltségén a

$$K(x) = \sum_{(i,j) \in E} x_{ij} k_{ij}(x_{ij}) \tag{1.4}$$

mennyiséget értjük.

A P_{st} út szállítási költségét az alábbi összefüggés adja meg:

$$K(P_{st}) = \sum_{(i,j) \in P_{st}} x_{ij}^{st} k_{ij}(x_{ij}).$$

Nyilván fennáll a következő összefüggés:

$$K(x) = \sum_{\substack{s \in S \\ t \in T}} K(P_{st})$$

és

$$\sum_{(i,j) \in E} x_{ij} = \sum_{(i,j) \in E} \sum_{\substack{s \in S \\ t \in T}} x_{ij}^{st}$$

A forgalomelosztási feladat adott hálózati és forgalmi adatok alapján olyan lehetséges folyamot meghatározni, amely bizonyos szempontból legjobban lefedi a forgalmi szituációt. A forgalmi helyzetet két szempontból vizsgálhatjuk: az egyéni utazási szokások és az összes utazási költség (vagy ami ezzel azonos az egy utazásra eső átlagköltség) szempontjából. WARDROP [68] két elvet fektetett le a fentiekkel kapcsolatban:

I. Az utazás az $s \in S$ és $t \in T$ pontok között olyan utakon bonyolódik le, amelyek utazási költsége kisebb vagy egyenlő a pontokat összekötő, de utazásra fel nem használt utaknál.

II. Az összes utazás összköltsége minimális.

Azon legkisebb költségű megoldást, amely az I. *Wardrop elven* nyugszik, egyéni a II. elv alapján kapottat pedig rendszeroptimálisnak nevezzük. Az elnevezés nyil-

vánvaló, ugyanis az egyén szempontjából az a jó, ha olyan úton megy, amelynek költsége kisebb a még igénybe vehető utaknál. A társadalom viszont olyan irányítási politikát igyekszik megvalósítani, amely az összes utazást tekintve legkevesebbe kerül. Az egyéni optimális feladat matematikai megfogalmazását és a megoldási algoritmusokat a későbbiekben tárgyaljuk. A feladat egy x megoldása eleget tesz az (1.2), (1.3) feltételeknek, és a *Wardrop I. elvének.*

A rendszeroptimális feladatot a következőképp fogalmazzuk meg: határozzuk meg azt az (1.2) és 1.3)-nak eleget tevő $x=(x_{ij})$ lehetséges megoldást, amelyre (1.4) minimális. Az (1.2)—(1.4) feladat egy több árucikkes maximális folyam feladat nemlineáris célfüggvénnyel.

1.2. *Összefüggés a két feladat megoldása között*

A fejezetben az egyéni és rendszeroptimális feladatok megoldásai közötti összefüggést vizsgáljuk meg. A két feladat optimális megoldása a legtöbb feladattípusnál nem esik egybe. A megoldások közötti kapcsolatot speciális feladatok esetén PIGOU [53] már 1920-ban vizsgálta.

WARDROP [68] egy forrás és nyelőpár és az őket összekötő n út esetén megmutatta, hogy a két megoldás nem egyezik meg.

Először tekintsük azt az egyszerű esetet, amikor egy s forrás és egy t nyelő adott, és az éleknek nincs kapacitása és legyen $s=1$, $t=n$. Jelölje $\mathbf{A}=(a_{ij})$ a digráfhoz tartozó incidencia mátrixot, amelynek i-edik sora a digráf egy pontjához és j-edik oszlopa a digráf egy (p, q) éléhez tartozik, az a_{ij} eleme pedig a következő:

$$a_{ij} = \begin{cases} 1, & \text{ha} \quad i = p, \\ -1, & \text{ha} \quad i = q, \\ 0, & \text{egyébként.} \end{cases}$$

A rendszeroptimális feladat a következőképp fogalmazható meg: határozzuk meg azt az s pontból a t pontba vezető v értékű x folyamot, amelyre:

$$\sum_{j=1}^{m} a_{ij} x_{ij} = \begin{cases} v, & \text{ha} \quad i = s, \\ -v, & \text{ha} \quad i = t, \\ 0, & \text{egyébként} \end{cases}$$

$$x_{ij} \geqq 0,$$

és

$$\sum_{(i,k)} x_{ij} k_{ij} \to \min,$$

ahol x_{ij} a folyam értéke, és $k_{ij} \geqq 0$ az egységnyi szállítási költség az (i, j) élen. Mátrix formában felírva az alábbi primál feladatra jutunk:

$$\begin{aligned} \mathbf{Ax} &= \mathbf{w} \\ \mathbf{x} &\geqq \mathbf{0} \\ \mathbf{kx} &\to \min, \end{aligned} \tag{1.5}$$

ahol $\mathbf{k}$ a szállítási költségek vektora és $\mathbf{w}=(v, 0, 0 \ldots 0, -v)$.

Ez egy egyszerű lineáris programozási feladat. Célszerűtlen azonban így megoldani, mivel a feladat duálja egy egyszerűbb algoritmushoz vezet. A duál feladat megfogalmazásához a $-\mu_i$ duál változókat használjuk μ_i helyett, hogy a megfelelő duális feladatot kapjuk.

Az (1.5) feladathoz tartozó duális feladat a következő: határozzuk meg a $-\boldsymbol{\mu} = (-\mu_1, -\mu_2, \ldots, -\mu_n)$ változókat, amelyre:

$$\sum_{i=1}^{m} -\mu_i a_{ij} \leqq k_j, \quad j = 1, 2, \ldots, m$$

$-\mu$-re nincs előjelmegkötés

és

$$-\sum_i \mu_i w_i \to \max,$$

vagy tömören

$$-\boldsymbol{\mu}\mathbf{A} \leqq \mathbf{k}$$

(1.6) $-\mu_i$-re nincs előjelmegkötés

$$-\boldsymbol{\mu}\mathbf{w} \to \max$$

Részletesen felírva a duál feladatot, az alábbi egyszerű formát kapjuk:

μ_i-re nincs előjel megkötés

(1.7) $$\mu_j - \mu_i \leqq k_{ij}, \quad (k, j) \in E$$

$$v(\mu_s - \mu_t) \to \max.$$

Ez utóbbi feladat éppen egy legrövidebb út feladat. Megoldása eleget tesz az egyéni optimális elvnek, mivel a közlekedésre igénybe vett utak közül a legkisebb költségűt szolgáltatja.

A dualitási tétel miatt, ha a két feladatnak van megoldása, akkor van optimális megoldása is, és az optimális megoldásához tartozó célfüggvény értéke megyezik. Az egyensúlyi tétel szerint, ha $\bar{\boldsymbol{\mu}}$ és $\bar{\mathbf{x}}$ a duál, illetve a primál feladat optimális megoldása, akkor:

$$\text{ha} \quad \bar{\mu}_j - \bar{\mu}_i < k_{ij}, \quad \text{akkor} \quad \bar{x}_{ij} = 0$$

és

$$\text{ha} \quad \bar{x}_{ij} > 0, \quad \text{akkor} \quad \bar{\mu}_j - \bar{\mu}_i = k_{ij}.$$

Így olyan élen, amelyik a legrövidebb úthoz tartozik, a folyam értéke nem nulla; fordítva viszont, ha valamely él nincs a legrövidebb útban, akkor az élen levő folyam értéke nulla.

A fentiekből következik, hogy ebben a speciális esetben az egyéni optimális és rendszeroptimális feladat megoldása éppen megegyezik, így a két feladat ekvivalens.

Ha a hálózat kapacitásos, akkor a következő primál feladathoz jutunk:

$$\mathbf{A}\mathbf{x} = \mathbf{w}$$

$$\mathbf{x} \leqq \mathbf{b}$$

(1.8) $$\mathbf{x} \geqq \mathbf{O}$$

$$\mathbf{k}\mathbf{x} \to \min,$$

ahol **x**, **w**, **k** a fent elmondott vektorok, **b** pedig az élkapacitások vektora. Ez egy előírt folyamértéken levő minimális költséges folyamfeladat.

Az (1.8)-hoz tartozó duális feladat a következő:

$$(-\boldsymbol{\mu}, -\boldsymbol{\lambda})\begin{pmatrix}\mathbf{A}\\ \mathbf{Z}\end{pmatrix} \leqq \mathbf{k}$$

(1.9)
$$-\boldsymbol{\lambda} \text{ nincs előjelben korlátozva}$$
$$-\boldsymbol{\mu} \geqq O$$
$$-\boldsymbol{\lambda}\mathbf{w} - \boldsymbol{\mu}\mathbf{b} \to \max,$$

ahol $-\boldsymbol{\lambda}$, $-\boldsymbol{\mu}$ duális változók vektorai, **Z** pedig $m \times m$-es egységmátrix. A feladat mátrixa igen sok nulla elemet tartalmaz.

Az (1.9) feladat a következő egyszerű alakra hozható:

$$\lambda_j - \lambda_i - \mu_{ij} \leqq k_{ij}, \quad (i,j) \in E$$
$$-\lambda_i\text{-re nincs előjelmegkötés}$$
(1.10)
$$-\mu_{ij} \geqq 0$$
$$v(\lambda_t - \lambda_s) - \sum \mu_{ij} b_{ij} \to \max,$$

ahol $-\lambda_i$, $i \in N$ a pontokhoz $-\mu_{ij}$, $(i,j) \in E$ az élekhez rendelt duális változók.

A dualitási tétel miatt optimális $\bar{x}_{ij}$, $\bar{\lambda}_i$, $\bar{\mu}_{ij}$ megoldás esetén a két célfüggvény megegyezik, azaz:

(1.11)
$$\sum_{i,j} k_{ij}\bar{x}_{ij} = v(\bar{\lambda}_t - \bar{\lambda}_s) - \sum_{i,j} \bar{\mu}_{ij} b_{ij}.$$

Az egyensúlyi tétel miatt a következő feltételek teljesülnek:

(1.12) — ha $\bar{\lambda}_j - \bar{\lambda}_i - \bar{\mu}_{ij} < k_{ij}$, akkor $\bar{x}_{ij} = 0$,

(1.13) — ha $\bar{\mu}_{ij} < 0$, akkor $\bar{x}_{ij} = b_{ij}$,

(1.14) — ha $\bar{x}_{ij} < 0$, akkor $\bar{\lambda}_j - \bar{\lambda}_i - \bar{\mu}_{ij} = k_{ij}$,

(1.15) — ha $\bar{x}_{ij} < b_{ij}$, akkor $\bar{\mu}_{ij} = 0$.

Az utóbbi két összefüggést összevonva kapjuk, hogy ha

(1.16)
$$0 \leqq \bar{x}_{ij} \leqq b_{ij}, \quad \text{akkor} \quad \bar{\lambda}_j - \bar{\lambda}_i = k_{ij}.$$

A fentiekből sejthető, hogy λ_i a legrövidebb útnál megszokott potenciál jellegű mennyiség. A μ_{ij} az (i,j) él szabad kapacitásával összefüggő változó: ha a folyam az élen pozitív, de a kapacitáskorlát alatt marad, akkor nulla, egyébként, ha nagyobb nullánál, akkor a folyamérték az él kapacitásával lesz egyenlő. Egy (i,j) él telített, ha $x_{ij} = b_{ij}$, egyébként telítetlen. Egy x folyam esetén, ha az s-ből t-be vezető P_1 útról a rajta menő folyam egy részét áttehetjük egy olyan s-ből a t-be vezető P_2 útra, amelynek az összes nem közös éle telítetlen, akkor a P_2 utat a P_1-re nézve lehetségesnek nevezzük, egyébként lehetetlen.

Az egyéni optimális feladatot ennek alapján átfogalmazhatjuk a következőképp:

I. Az utas olyan útvonalakat választ, amelyek költsége kisebb, mint bármelyik olyan útvonalé, amely ezekre nézve lehetséges.

A fenti átfogalmazást és a primális és duál feladatot felhasználva megmutatjuk, hogy a rendszeroptimális feladat megoldása ebben az esetben is megoldása az egyéni optimális feladatnak.

Tegyük fel, hogy P_1 egy olyan út, amelyen az optimális $\bar{x}_{ij}$ folyam egy része megy.

Erre az útra és az úthoz tartozó $\sum k_{ij}$, $(i,j)\in P_1$ utazási egységköltségre érvényes a következő összefüggés:

$$\sum_{P_1}(\bar{\lambda}_j-\bar{\lambda}_i-\bar{\mu}_{ij})=\sum_{P_1}k_{ij}, \tag{1.17}$$

azaz

$$\bar{\lambda}_t-\sum_{P_1}\bar{\mu}_{ij}=\sum_{P_1}k_{ij}, \tag{1.18}$$

ahol az összegzés a P_1 úthoz tartozó (i,j) élre történik, és λ_s-t nullának választottuk.

Egy másik, P_1-re nézve lehetséges P_2 utat választva kapjuk (1.15)-ből:

$$\sum_{P_1}\bar{\mu}_{ij}\geqq\sum_{P_2}\bar{\mu}_{ij}. \tag{1.19}$$

Az (1.18)-at és (1.19)-et összevetve kapjuk:

$$\sum_{P_1}k_{ij}=\bar{\lambda}_t-\sum\bar{\mu}_{ij}\leqq\bar{\lambda}_t-\sum_{P_2}\bar{\mu}_{ij}\leqq\sum_{P_2}k_{ij}.$$

Azaz az optimális megoldáshoz nem tartozó P_2 útra áttéve a forgalmat egy, a megoldáshoz tartozó P_1 útról, az egyéni utazás szempontjából is előnytelen utat kaphatunk. Így a rendszer optimális feladat megoldása egyben egyéni optimális megoldást is szolgáltat.

Visszafelé ez az állítás nem igaz, azaz az egyéni optimális feladat nem minden megoldása szolgáltat egyben rendszeroptimális megoldást is (lásd MANDL) [43]).

A továbbiakban az általános esettel foglalkozunk, amit az 1.1 pontban fogalmaztunk meg. Adott tehát S források és T nyelők halmaza és az s, $s\in S$ pontból a t, $t\in T$ pontba h_{st} folyamérték folyik.

Az $x_{ij}k_{ij}(x_{ij})$ költségfüggvényről feltesszük, hogy növekvő és szigorúan konvex. Ennek megfelelően a $k_{ij}(x_{ij})$ függvény pozitív és szigorúan növekvő. Először az egyéni optimális feladatot fogalmazzuk át. Tekintsünk egy rögzített s, t forrás-nyelő párt, és az s pontból a t pontba vezető P_1 és P_2 utat. Jelölje az (i,j), $(i,j)\in P_1$ élen az s-ből a t-be menő folyamot $r=x_{ij1}$ az (i,j), $(i,j)\in P_2$ élen pedig $q=x_{ij2}$. Tegyünk át a P_1 útról a P_2 útra $z\leqq r$ folyamot.

Legyen az új folyam értéke tehát a következő:

$$\begin{aligned}\bar{x}_{ij1}&=x_{ij1}-z>0, \quad (i,j)\in P_1\\ \bar{x}_{ij2}&=x_{ij2}+z>0, \quad (i,j)\in P_2.\end{aligned} \tag{1.20}$$

A P_1 és P_2 út egységnyi utazási költsége

$$k\big(P_1(\bar{x})\big)=\sum_{(i,j)\in P_1}k_{ij}(\bar{x}_{ij}) \tag{1.21}$$

$$k\big(P_2(\bar{x})\big)=\sum_{(i,j)\in P_2}k_{ij}(\bar{x}_{ij}). \tag{1.22}$$

A folyam értéke (1.20) miatt az egész hálózaton megváltozott:

$$\bar{x}_{ij} = \begin{cases} x_{ij} - z, & \text{ha} \quad (i,j) \in P_1, \quad (i,j) \notin P_2 \\ x_{ij} + z, & \text{ha} \quad (i,j) \in P_2, \quad (i,j) \notin P_1 \\ x_{ij}, & \text{egyébként.} \end{cases} \tag{1.23}$$

Az egyéni optimális feladatot az (1.20)—(1.23) felhasználásával a következőképp fogalmazzuk át:

Az x folyam az egyéni optimális feladat megoldása, ha minden $s \in S$ és $t \in T$ párra, minden s-ből a t-be vezető P_i és P_j útra, és minden lehetséges $z > 0$-ra fennáll a következő összefüggés:

$$k(P_i(x)) \leqq k(P_j(\bar{x})). \tag{1.24}$$

Más szavakkal ez azt jelenti, hogy x egyéni optimális megoldás esetén a folyam átcsoportosítása bármely közlekedésre igénybevett útról egy másik útra az (1.24)-ben megfogalmazott költségnövekedést eredményezhet.

A következő tétel az egyéni optimális feladat megoldásával kapcsolatos.

1.1. TÉTEL: Egy x megengedett megoldás akkor és csakis akkor megoldása az egyéni optimális feladatnak, ha minden $s \in S$, $t \in T$ forrás-nyelő párra létezik az s-ből a t-be vezető utak egy olyan költség szerint csökkenő sorbarendezése, amelyre

$$k(P_1) = k(P_2) = \ldots = k(P_q) \leqq k(P_{q+1}) \leqq \ldots \leqq k(P_r), \tag{1.25}$$

és az x folyam ezekre az utakra

$$x_{ij}^{st} \begin{cases} > 0, & \text{ha} \quad (i,j) \in P_i, \quad i = 1, 2, \ldots, q, \\ = 0, & \text{egyébként.} \end{cases} \tag{1.26}$$

Más szavakkal az utak rögzített s, t pár esetén a H_1 és H_2 csoportra bonthatók. A $P_i \in H_1$ utakon megy folyam, a $P_j \in H_2$ utakon a folyam értéke nulla. A $P_i \in H_1$ utak költsége egyenlő, de kisebb egyenlő a $P_j \in H_2$ utak költségénél.

Bizonyítás

Először megmutatjuk, hogy a fenti feltétel elegendő, azaz bármely $z > 0$ megengedett folyamérték-átrendelés egy $P_1 \in H_1$ útról egy $P_2 \in H_2$ útra az (1.24) összefüggést eredményezi.

A P_1 és P_2 út ilyen választása mellett

$$k(P_1(x)) \leqq k(P_2(x)) \tag{1.27}$$

feltétel teljesül.

A teljes élköltség szigorúan konvex, így $\bar{x}_{ij} > x_{ij}$ esetén

$$k(\bar{x}_{ij}) \geqq k(x_{ij}).$$

Mivel egy lehetséges (1.20) átrendelés legalább egy (i,j), $(i,j) \in P_2$ élen folyamérték-növekedést okoz, így fennáll a következő összefüggés:

$$\sum_{P_2} k_{ij}(\bar{x}_{ij}) = k(P_2(\bar{x})) > k(P_2(x)) = \sum_{P_2} k_{ij}(x_{ij}). \tag{1.28}$$

Az (1.27) és (1.28) feltételeket összevetve kapjuk, hogy

$$k(P_1(x)) < k(P_2(\bar{x}))$$

ami kielégíti az (1.24) feltételt.

Megmutatjuk, hogy az (1.25), (1.26) feltétel szükséges az egyéni optimális feladat megoldásához. Tételezzük fel, hogy van olyan $P_1 \in H_1$ és $P_2 \in H_2$ út, amelyre

$$\begin{aligned} &k(P_1(x)) < k(P_2(x)), \\ &\text{és} \\ &x_{ij}^{(1)} > 0. \end{aligned} \tag{1.29}$$

Megmutatjuk, hogy az x megoldás ebben az esetben nem egyéni optimális. A feltételezés miatt az $x_{ij}k_{ij}(x_{ij})$ függvény szigorúan konvex, növekvő így a $k_{ij}(x_{ij})$ függvény folytonos. Így megadható egy olyan lehetséges z folyamátrendelés a P_1 útról a P_2 útra, amelyre teljesül a következő egyenlőtlenség:

$$k(P_1(x)) > k(P_1(\bar{x})) > k(P_2(x)). \tag{1.30}$$

A fenti feltétel viszont ellentmond az (1.24) definíciónak, így az (1.29) feltételezés hibás.

Vizsgáljuk meg a rendszeroptimális feladat megoldását is általános esetben. Ehhez az úton és az élen folyó folyam ún. marginális költségét definiáljuk. A marginális költség a folyam megváltoztatásának hatására bekövetkezett költségváltozás, azaz az $x_{ij}k(x_{ij})$ függvény deriváltja.

Az (i, j) él marginális költsége a következő:

$$d_{ij}(x_{ij}) = \frac{dx_{ij}k_{ij}(x_{ij})}{dx_{ij}}. \tag{1.31}$$

Egy út marginális költsége az úthoz tartozó élek marginális költségének összege, azaz

$$D(P_i(x)) = \sum_{(i,j)\in P_i} d_{ij}(x_{ij}). \tag{1.32}$$

Mivel $x_{ij}k_{ij}(x_{ij})$ szigorúan konvex, növekedő, így az (1.31) és (1.32) marginális költségfüggvény pozitív, folytonos és növekvő. A rendszeroptimális feladat $K(x)$ célfüggvényét (1.4)-ben adtuk meg. A következő tétel a rendszeroptimális feladat x megoldására ad szükséges és elegendő feltételt az (1.31), (1.32) marginális költségek felhasználásával.

1.2. TÉTEL: Az x folyam akkor és csakis akkor megoldása a rendszeroptimális feladatnak, ha minden s, t forrás-nyelő párra megadható az s-ből a t-be vezrtő utak egy olyan $1, 2, .. q, q, +1, \ldots, r$ sorbarendezése, amelyre

$$\begin{aligned} D(P_1(x)) = D(P_2(x)) = \ldots = D(P_q(x)) \leqq D(P_{q+1}(x)) \leqq \\ \leqq D(P_{q+2}(x)) \leqq \ldots \leqq D(P_r(x)) \end{aligned} \tag{1.33}$$

és

$$x_{ij}^{st} \begin{cases} > 0, & \text{ha} \quad (i,j) \in P_i, \quad i = 1, 2, \ldots, q, \\ = 0, & \text{egyébként.} \end{cases} \tag{1.34}$$

A tételt nem bizonyítjuk, mivel bizonyítása hosszadalmas. Részletes bizonyítást POTTS—OLIVER ad [57] könyvében.

Összevetve az egyéni és rendszeroptimális feladat megoldását azt kapjuk, hogy mindkét megoldás a forrás-nyelő párok közötti utak esetén eleget tesz bizonyos költség feltételeknek. Nevezetesen az egyéni optimális feladat megoldása során kapott megoldás egyenlő költségű utakon bonyolítja le a forgalmat.

A rendszeroptimális feladat megoldására hasonló igaz, de az utak költsége helyett azok marginális költségét kell venni.

A fenti két tétel lehetőséget ad a két megoldás közötti összehasonlításra. Megmutatjuk, hogy mely esetben lesz a két megoldás egyenlő, konvex célfüggvény esetén.

Tekintsük a következő $\bar{k}_{ij}(x_{ij})$ függvényt:

$$\bar{k}_{ij}(x_{ij}) = \frac{1}{x_{ij}} \int_0^{x_{ij}} k_{ij}(y)\, dy. \tag{1.35}$$

A $\bar{k}_{ij}(x_{ij})$ függvényt költségfüggvénynek tekintve, a marginális költségüggvény a következő:

$$\bar{d}_{ij}(x_{ij}) = \frac{dx_{ij}\bar{k}(x_{ij})}{dx_{ij}} = k_{ij}(x_{ij}).$$

Ez azt jelenti, hogy ha a $\bar{k}_{ij}(x_{ij})$ függvényt alkalmazzuk a rendszeroptimális feladatban, akkor az így kapott $\bar{x}$ optimális megoldás egyben megoldása az egyéni optimális feladatnak is, azaz

$$k(P_i(\bar{x})) = \bar{D}(P_i(\bar{x})).$$

1.3. *Integrált elosztási modellek*

Az 1.1 pontban feltettük, hogy a $\mathbf{H}=(h_{st})$ mátrix rögzített. Gyakran felvetődik az a probléma, hogy mi történik, ha csak a forrásból kiinduló, vagy csak a nyelőbe beérkező mennyiségeket rögzítjük le, vagy esetleg csak az összes utazás számát. Ez a tényhelyzet és az idealizált elképzelések közötti különbség vizsgálatára, esetleges csökkentésére adhat lehetőséget.

Mint látni fogjuk, az alternatív átfogalmazás akár az egész tervezési eljárás átalatására is lehetőséget ad. Ennek fő előnye elsősorban abban mutatkozik meg, hogy az egész eljárás során azonos hálózati és egyéb paraméterekkel dolgozunk. Így például nem lesz más a költségfüggvény az előrebecslés és ráterhelés folyamán, aminek előnyét nem kell bizonyítani.

Az alábbiakban összefoglaljuk a főbb modelleket:

a1. Az utazók kiindulási és célállomása kötött, az útvonalukat azonban szabadon választhatják meg. Ez a klasszikus forgalomelosztási feladat, amelynél adott a $\mathbf{H}=(h_{ij})$, $\mathbf{O}=(o_i)$, $\mathbf{D}=(d_j)$, meghatározandó az $\mathbf{X}=(x_{ij})$ forgalom.

a2. Az utazók kiindulási körzete rögzített végállomásukat és útvonalukat szabadon megválaszthatják. Ez egy kombinált nyelő-orientált forgalomszétosztási és elosztási modell, ahol rögzített az $\mathbf{O}=(o_i)$ vektor, meghatározandó az $\mathbf{X}=(x_{ij})$ (és ami adódik a $\mathbf{H}=(h_{ij})$) és a $\mathbf{D}=(d_j)$ vektor.

a3. Az utazók célkörzete rögzített, kiindulási körzetüket és útvonalukat szabadon választhatják. A feladat forrás-orientált forgalomszétosztási és elosztási modell, ahol rögzített a $\mathbf{D}=(d_j)$ vektor, meghatározandó $\mathbf{X}=(x_{ij})$, valamint $\mathbf{H}=(h_{ij})$ és $\mathbf{O}=(o_i)$.

a4. Az utazók szabadon választhatják kiindulási és célkörzetüket. A modell forgalomkeltési, forgalomszétosztási és forgalomelosztási feladatok kombinációja. Rögzített az utazások száma, w, meghatározandó $\mathbf{X}=(x_{ij})$, $\mathbf{O}=(o_i)$, $\mathbf{D}=(d_j)$ és $\mathbf{H}=(h_{ij})$.

Megmutatjuk, hogy a feladatok mindegyike átfogalmazható az a1. feladatra.

Tekintsük az a2 problémát. Az eredeti hálózat N pontjainak halmazát bővítsük egy fiktív q ponttal, és az E halmazhoz pedig vegyük hozzá a (t, q) éleket $(t \in T)$. Legyen a szállítási költség az új éleken nulla, azaz $k_{tq}(x_{tq})=0$ és $b_{tq}=\infty$. Az így kapott (N', E') irányított gráfban a forrásokra megfogalmazott feltételek ((1.3)-ban) teljesülnek. A nyelőpontokra külön feltételek nincsenek előírva, csak a q pontra. Erre viszont a közbülső pontokra előírt feltételek miatt teljesül a

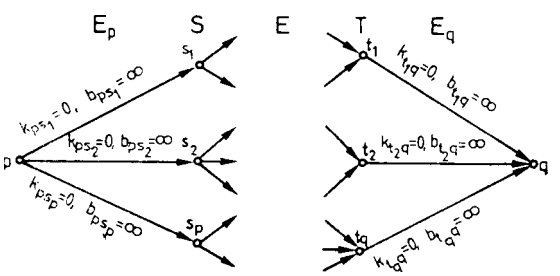
$$\sum_{t,q} x_{tq} - \sum_{t,q} x_{qt} = \sum_{t,q} x_{tq} = d_q = w$$

feltétel. Akár rendszer optimális, akár egyéni optimális feladatként oldjuk meg a feladatot, a $t \in T$ pontokra kiadódó

$$d_t = \sum_{p,t} x_{pt} - \sum_{t,q} x_{tq}$$

érték szintén optimális folyamértékekből adódik. Ugyanis a megoldás szempontjából a $k_{tq}(x_{tq})$ és a b_{tq} értékek közömbösek. Az elmondott eljáráshoz hasonlóan fogalmazhatjuk át az a3 feladatot is az a1 feladatra.

Az a4 feladat átfogalmazását az alábbiakban adjuk meg. Vegyünk fel egy fiktív p és q pontot, és legyen $N'=(N \cup p \cup q)$. Az élek halmazát is bővítjük. A p pontot kössük össze az $s \in S$ forráspontokkal és jelöljük az így kapott irányított élhalmazt E_p-vel. Hasonlóképp kössük össze a $t \in T$ nyelőpontokat a q ponttal és a kapott élhalmazt jelöljük E_q-val. Legyen $E'=(E \cup E_p \cup E_q)$, $k_{ps}(x_{ps})=k_{tq}(x_{tq})=0$, $b_{ps}=b_{tq}=\infty$ és $l_{ps}=l_{tq}=0$. Határozzuk meg az így kapott (N', E', l, k, b) közlekedési hálózaton az $x=(x_{ij})$ folyamot, amely egyéni vagy rendszeroptimális. Az a2 feladatnál elmondott okoskodás alapján könnyen belátható, hogy az így kapott megoldás az eredeti (N, E, l, k, b) hálózaton is egyéni vagy rendszeroptimális.



1. ábra
Az a4 feladat átfogalmazása az a1 feladatra

Megjegyezzük, hogy az a1 feladat és az a2, a3, a4 feladat megoldása rendszerint nem lesz azonos. Ugyanis az a1 feladatban rögzített $d_1, d_2, \ldots d_s$ értékek rendszerint nem azonosak az a2 feladatban kapott $\bar{d}_1, \bar{d}_2, \ldots, \bar{d}_s$ értékekkel, mivel az eredeti d_i értékeket nem az itt alkalmazott elvekből kapjuk, hanem a forgalomkeltési modell eredményeképp vagy valamilyen egyéb megfontolásból. Ráadásul a h_{st} értékeket is nem az egyéni vagy rendszeroptimális megoldás eredményeképp kaptuk, hanem egészen más célfüggvényből.

A már említett felhasználási lehetőségek mellett néhány további fontos területet említünk meg az a2—a4 modellekkel kapcsolatban:

— a2 modell a jelenlegi (jövőbeni) úthálózat alapján az utazási céltól függően javaslatot ad a „végcélok" telepítési helyeire. Nevezetesen munkahelyi utazások esetén a hálózati paraméterek figyelembevételével munkahelyek telepítésének helyére kapunk ötleteket. Hasonló eredményeket kapunk hétvégi szórakozási lehetőségek elhelyezésére vagy más típusú utazási célokra is.

— a3 modell annak a tanulmányozására ad lehetőséget, hogy a célkörzet utazásai — figyelembe véve az úthálózat paramétereit — milyen induló körzetekből tevődnek össze ideális esetben. Ez pedig például lakás—munkahely forgalom esetén a lakókörzetek fejlesztési irányaira ad ötleteket.

— a4 modell az egész tervezési eljárást is megmódosítja — amennyiben magába foglalja a forgalomkeltés, szétosztás és elosztás modelljeit. Ha az útszakaszok beruházási költségét is a modellbe tesszük, úgy egyetlen eljárással helyettesíthetők az eddigi modellek jó része.

A gondolat nem új. Számos szerző dolgozott ki ilyen típusú kombinált modelleket (EVANS [21], ERLANDER [20], FLORIAN—NGUYEN [23], HOLM—JENSEN—NIELSEN [30], POTTS—OLIVER [57]). A részletes tárgyalása meghaladja a dolgozat méreteit.

1.4. *Megoldási algoritmusok*

Mint az előző fejezetekben láttuk, mind a *Wardrop I* mind *II. elve* alapján a költségfüggvénytől függően más feladatot kapunk. A feladat megoldása így a feladat kitűzésétől függ. Lineáris költségfüggvény és kapacitáskorlát nélküli hálózatoknál a rendszeroptimális feladat lineáris programozási feladatot eredményez. Duálisa viszont legrövidebb út feladat, ezért előnyösebb ezt a feladatot megoldani. A későbbiekben ismertetendő ún. minden — vagy semmit algoritmus éppen ebben az esetben adja meg a feladat megoldását. Kapacitáskorlátos és lineáris költségfüggvény esetén egyéni felső korlátos lineáris programozási feladatot kell megoldanunk. Az így megfogalmazott feladat viszont nagyon nagy méretű, és még kis pont és élszám esetén is tetemes futási időt igényel.

A feladat viszont megfogalmazható több árucikkes maximális folyam problémaként is, amelynek célfüggvénye lineáris. Ezt vagy potenciál módszerrel (l. KLAFSZKY [38]) vagy az *out-of-kilter eljárással* (FULKERSON [27]) oldhatjuk meg.

Ezen módszerek számítástechnikai szempontból is értékelt összefoglalásai nem régen jelentek meg (ASSAD [2], KENNINGTON [36](. MANDL [43] könyvében a KLEIN [39] módszert javasolja ilyen feladatok esetén. TOMLIN [60] *Dantzig—Wolfe dekompozíciós módszerrel* oldja meg a feladatot.

Konvex célfüggvény esetén egy konvex programozási feladatot kell megoldani (CHARNES—COOPER [13]). Az egyéni optimális feladat megoldására HALL—PETTERSON [29] ebben az esetben geometriai programozást használ, és a duális feladatot oldja meg. DAFERMOS—SPARROW [14] egy *Taylor-sor* közelítést alkalmaz algoritmusában. NGUEN [51] először a konvex programozási feladat dekompozíciójával dolgozik. Későbbi cikkében (l. [52]) egy speciális feladatra ZOUTENDIJK [71] egyik eljárását alkalmazza. Eljárásának számítástechnikai tapasztlatairól számol be RUITER [58]. IRI [33] Japán közúti hálózatának tervezéséhez büntetőfüggvényt használ.

Eddig az egyéni optimális feladat esetén feltettük, hogy az utas a *Wardrop I. elvének* megfelelően közlekedik. Az egyén utazási szokásainak tanulmányozása során azonban rájöttek, hogy az utas a gazdaságossági megfontolások mellett egy sor egyéb szempontot is figyelembe vesz (kényelem, táj stb.) útvonalának megválasztása során (l. például HOROWITZ [31]). Így az utasok egy része nem a leggazdaságosabb útvonalat veszi igénybe (MICHAELS [44], WACHS [67]). Ezért célszerű a leggazdaságosabb útvonal mellett egyéb alternatív (esetleg 2., 3., ... leggazdaságosabb) útvonalakat is figyelembe vevő modelleket kidolgozni. Erre a fejezet későbbi részében még visszatérünk. Néhány szerző ennek figyelembevételére sztochasztikus elemeket is tartalmazó eljárást javasol (BURELL [12], VLIET [66], WHITMAN [70]).

A fent ismertetett optimalizációs eljárások matematikai szempontból elegánsak, és a kiinduló hipotézisnek eleget tevő megoldást szolgáltatnak. Néhány ezer csomópont és él esetén azonban, a futási idők és a tárolókapacitás tetemes volta miatt nem megfelelőek. IRI [33] egy 5000 pontból és 2000 élből álló feladat esetén megmutatja, hogy matematikai programozási módszert választva 11 ezer változóból álló feladatot kapunk, míg az 1.4 pontban ismertetendő eljáráshoz összesen 40 ezer tárolóhely szükséges. A futási időt tekintve ez utóbbi jóval rövidebb. Egyéni optimális feladat megoldására ezért a legtöbb szerző heurisztikus módszert javasolt. Ezek — megfelelő lépésközt használva közel vannak az optimumhoz, és nagy méretek esetén is jól alkalmazhatók.

Ilyen eljárással dolgoznak a számítógépgyártó cégek és software házak által készített programcsomagok (ADECODE [1], UTP [62], VENUS [63], ICL programcsomag — lásd BÉNYEI—PARLAGI [10], SYSTAN [70], TRANPLAN [61], FHWA [22]).

A hazai szerzők is heurisztikus technikát használnak a forgalomelosztási feladat megoldására (NAGY K.—MARTON M. [50], MONIGL [46], BAKÓ [5].

A továbbiakban összefoglaljuk a fontosabb alapmódszereket. A modellek részletes ismertetésével, elemzésével számos szerző foglalkozott: BAKÓ [8], BURELL [12], LEBLANK [41], MORRIS [48], POTTS—OLIVER [57], MANDL [43]).

1.4.1. *Minden vagy semmi algoritmus*

A legegyszerűbb és a legkorábban használt eljárás a forgalomelosztásra azon a hipotézisen alapul, hogy a forgalom két pont között egy — a leggazdaságosabb útvonalon bonyolódik le. Az eljárásban ezért elegendő m lépésben kiszámolni a minimális kifeszítő fát, és az így kapott utakhoz rendeljük hozzá a megfelelő h_{ij} elemeket. A feladatot megoldó E1 algoritmus a következő lépésekből áll:

E10: Legyen $x_{ij}=0$, $a=1, f=s_1, g=t_1$.

E11: Határozzuk meg az $f=s_a$ pontból a digráf összes $t\in T$ pontot tartalmazó minimális költségű kifeszítő fát, $b=1, f=s_a$.

E12: A P_{fg} úton növeljük a forgalmat h_{fg}-vel: $x_{ij}=x_{ij}+h_{fg}$, $(i,j)\in P_{fg}$.

E13: Ha $b\leqq q$, legyen $b=b+1$, $g=t_b$ és menjünk E12-re, egyébként E14-re.

E14: Ha $a\leqq p$, legyen $a=a+1$ és menjünk E11-re, egyébként E15-re.

E15: STOP.

A fenti eljárás önmagában kevésbé alkalmazható, de — mint a későbbiekben látni fogjuk — részfeladatként a legtöbb eljárás felhasználja.
Az E1 algoritmussal kapcsolatban a következő problémák merülnek fel:

— Az alaphipotézis hibás, mivel az utasok jó része különböző megfontolások miatt nem a legrövidebb úton megy (MICHAELS [44], WACHS [67]).
— A leggazdaságosabb útvonal a kiszámítás után az első lépésben megváltozhat, mivel a kifeszítő fabeli utaknak közös részei is vannak, és az élköltség függ a rajta levő forgalomtól.
— A módszer instabil, mivel triviális élköltség változás az eredményben komoly változásokat okozhat (lásd DIAL [15]).
— Az eredményül kapott megoldáshoz tartozó utazási összköltség lényegesen kevesebb a valódi költségnél.
— Bizonyos élek forgalmát túlbecsüli, másokat — amelyeken a megoszló forgalom bonyolódna le — alulbecsülni.

Az eljárás egy iterációs változatát írja le MORRIS [48] összehasonlító munkájában. Az algoritmus lényege az, hogy minden fa meghatározása után a legrövidebb út kiszámolásánál figyelembe vesszük az élterheléseket, és az E1 algoritmus befejezése után az eljárást újra kezdjük a már az éleken levő forgalom figyelembevételével. Az egymást követő lépések során oszcillációs is felléphet (lásd FHA [22] III. fejeze-16. o.). Ennek kiküszöbölésére MORRIS [48] az utazási idők súlyozását javasolja:

$$\bar{k}_{ij}^{(i)}(x_{ij}) = qk_{ij}^{(i-1)}(x_{ij}^{(i-1)})+(1-q)k_{ij}^{(i)}(x_{ij}).$$

Ezt a súlyozást alkalmazva az eljárás konvergens lesz. A módszer kapacitáskorlátos változatával lényegesen pontosabb eredményeket kapunk. Ez a kettes és az ötös lépésben különbözik az eredeti eljárástól:

E20: Legyen $x_{ij}=0$, $a=1, f=s_1$, $g=t_1$, $y=0$

E21: Határozzuk meg az összes $t\in T$ pontot tartalmazó minimális költségű kifeszítő fát az $f=s_a$ pontból a $k_{ij}(x_{ij})$ költségfüggvénnyel és legyen $b=1$.

E22: A P_{fg} úton növeljük az x_{ij} értéket: ha $b_{ij}>x_{ij}+h_{fg}$, akkor $x_{ij}=x_{ij}+h_{fg}$, és $h_{fg}=0$, egyébként $r=x_{ij}$, $x_{ij}=b_{ij}$, $h_{fg}=h_{fg}-(b_{ij}-r)$, $y=1$, $t_{ij}=\infty$.

E23: Ha $b<q$, legyen $b=b+1$, $g=t_b$, menjünk E22-re, egyébként E24-re.

E24: Ha $a<p$, legyen $a=a+1$, menjünk E21-re, egyébként E25-re,

E25: Ha $y=1$, vegyük a nem nulla h_{ij} elemeket és végezzük el az E21—E24 lépéseket, egyébként folytassuk E26-ban.

E26: STOP.

Az E2 algoritmus éppen a felső korlátig terheli le az egyes éleket, amennyiben ezt eléri x_{ij}, más utat választ a forgalom lebonyolítására. Ha implicit felső korlátokat használunk (amelynél az utazási idő a kapacitás elérésekor igen nagy érték) az a helyzet állhat elő, hogy az algoritmus nem megfelelő értékeket ad azon forráspontokra, amelyekkel kezdődik az eljárás, de fokozatosan jó eredményt ad az élforgalmak növekedésével.

Összefoglalva a minden — vagy — semmi elv alapján készített eljárásokat rendszerint csak részfeladatként alkalmazhatjuk összetett algoritmusokban. Néhány esetben viszont alkalmazhatjuk a hálózat analízisére, szűk keresztmetszetek megállapítására, és közúti hálózat esetén több variáció elemzésére — a módszer egyszerűsége és gyorsasága miatt. Az összes lépések száma ugyanis igen csekély: m fa meghatározás és $m \cdot n \cdot k$ összeadás és összehasonlítás — ahol k a legrövidebb útban levő élek átlagos száma.

1.4.2. *Részenkénti forgalomelosztás*

Az egyéni optimális feladat megoldását abban az esetben lehetne egzakt módon megoldani, ha a forgalmat egységenként rendelnénk a hálózathoz, és arra az útra tennénk, amelyik az aktuális forgalmi szituációt figyelembe véve a leggazdaságosabb. Ez az eljárás az utazók nagy száma miatt (és az eljáráshoz szükséges tetemes gépidőigény miatt) nem járható. Ehelyett megpróbáljuk a forgalmat nagyobb egységekben hozzárendelni a hálózathoz, és ennek alapján kíséreljük meg a reális forgalmi szituációt megközelíteni.

Az algoritmus alapgondolata a következő: osszuk fel a $\mathbf{H}=(h_{ij})$ minden elemét részekre. Egy lépésben csak egy részt teszünk rá a hálózatra. Kiszámoljuk a hálózat utazási költségét a már élen levő forgalommal, majd a következő részt annak figyelembevételével rendeljük a hálózathoz. Az eljárást addig folytatjuk, ameddig az összes forgalmat el nem osztottuk a hálózaton. Tegyük fel, hogy adott egy $\mathbf{V}=(v_1, v_2 \ldots v_w)$ vektor, amelyre

$$\sum_{i=1}^{w} v_i = 1.$$

A feladatot az alábbi E3 algoritmussal oldjuk meg:

E30: Legyen $x_{ij}=0, r=1$.

E31: Legyen $i=1, j=1$.

E32: Határozzuk meg az s_i-ből kiinduló és az összes $t \in T$ pontot tartalmazó minimális költségű kifeszítő fát a $k_{ij}(x_{ij})$ költségfüggvénnyel.

E33: A $P_{s_i t_j}$ út (a, b) élén legyen $x_{ab} := x_{ab} + v_r h_{s_i t_j}, j=j+1$.

E34: Ha $j \leqq q$, menjünk E33-ra, egyébként $i=i+1$.
Ha $i \leqq p$, menjünk E32-re, egyébként menjünk E35-re.

E35: Legyen $r:=r+1$. Ha $r>w$, menjünk E36-ra, egyébként menjünk E31-re.

E36: STOP.

Az E3 algoritmus nagymértékben függ a $k_{ij}(x_{ij})$ függvény alakjától és az előre rögzített V vektortól. Néhány szerző egyenlő részekre való felosztást javasol. A hazai szerzők mind az egyenlő, mind a változó részekre való osztással dolgozó eljárásokat kidolgozták (BAKÓ [6], MONIGL—VÁSÁRHELYI [47]). VLIET [64] a londoni tervezési hálózat egy szűkített változatára tesztelte a különböző részekre való felosztást, és az általa választott célfüggvényre a négy egyenlő részre való osztást találta a legmegfelelőbbnek.

Az E3 algoritmusban, mint látjuk, a legproblematikusabb rész a V vektor meghatározása. Minél kisebb a v_i elemek értéke, annál pontosabb eredményt kapunk. A v_i értékek csökkentésével viszont tetemesen növekszik a futási idő. Ezért célszerű csak azokat az éleket kiválasztani, amelyek az egyes lépések után túl nagy költségfüggvényt adnak. A növekedés mértékét az $R^{(0)}=k_{ij}(x_{ij})/k_{ij}(0)$ érték adja az első és $R^{(r)}=k_{ij}(x_{ij}^{(r)})/k_{ij}(x_{ij}^{(r-1)})$ az r. lépésben. Az algoritmus során minden lépés után számoljuk ki az $R^{(r)}$ értéket, és azokat az éleket válasszuk ki, amelyekre $R^{(r)} \geqq \bar{R}$. Az $\bar{R}$ értékét gyakorlati tapasztalatok alapján VLIET [64] 4/3-ban adja meg. Minden kiválasztott élhez keressük meg a legnagyobb $h_{st}^{(r)}$ értéket, és végezzünk egy elosztási lépést az így kapott — $h_{st}^{(r)}$ mátrixszal és növeljük a H értékét a most visszavett értékkel. Ezután az eljárást az E32 lépésben folytassuk.

Egy további lehetséges módosítás az egy lépésben első és további alternatív utakra való forgalomterhelés. Ez csúcsidőben, forgalommal telített városokban jó eredményt szolgáltat. A k-adik legrövidebb út módszerek tárgyalása meghaladja a dolgozat méreteit. Ilyen eljárást használtunk a budapesti tömegközlekedés szétosztási feladatának megoldására. A kidolgozott eljárás gyors, és jó eredményeket szolgáltatott.

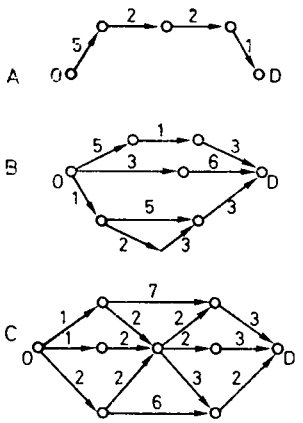
A részenkénti módszer előnye, hogy előre megadott lépésszám után befejeződik az eljárás. A matematikailag elegáns, és konvergenciát biztosító iterációs eljárások (EVANS [21], FLORIAN—NGUYEN [23], POTTS—OLIVER [57]), nagy méretek esetén túl nagy futási időket adnak, ezért inkább csak elméleti szempontból érdekesek.

Ezen módszerek programozása is komplikáltabb az eddig ismertetett eljárásoknál.

1.4.3. *Valószínűségi modellek*

Az eddig ismertetett módszereknél az utazási szokásokkal kapcsolatos feltételezésekkel szemben itt más, bizonyos valószínűségi megfontolásokat teszünk. Az ismertetendő eljárások alapötletei nem újak. Mi a különböző ötleteket összevetjük, rámutatunk az alkalmazási lehetőségekre és analizáljuk az előforduló problémákat.

Az ilyen elvvel dolgozó módszerek lényege az, hogy az utazásokat megpróbálják egyszerre vagy egymás utáni lépésekben több alternatív — nem biztos, hogy mindig a leggazdaságosabb — útszakaszokhoz hozzárendelni. Tekintsük a 2. ábrán megadott utakat az O forrás és D nyelőpont között:



2. ábra
Utazási lehetőségek két pont között

Az O pontból D pontba menő forgalom elosztására az A, B és C esetekben különféle lehetőségünk van. A 2. ábrán az A esetben egy lépésben egyetlen útra tesszük, míg a B esetben az alternatív legrövidebb utakon osztjuk el a forgalmat.

A C esetben bemutatott hálózat esetén a legrövidebb utak mellett a legrövidebb utakhoz távolságban (költségben) közel eső utakat is figyelembe veszünk a számoláskor. A legegyszerűbb eljárásoknál feltesszük, hogy a forgalom nagyságát előre nem tudja az utazó, ezért az eddig szerzett tapasztalatai alapján választ útvonalat.

Az egyén szempontjából figyelembe veendő $k_{ij}(x_{ij})$ függvény helyett ezért egy konstans k_{ij} értéket veszünk élenként, amelynek értékét a feltételezett forgalmi változások miatt a különböző modellekben változtatják. Egy él utazási ideje ugyanis a napi forgalom változásától függően — egyes szerzők szerint (WHITMANN [70], BURELL [11]) 10—20%-kal is eltérhet. Innen ered az ötlet, hogy variáljuk az útszakaszok utazási idejét bizonyos korlátok között. Az alapalgoritmusoktól ez a módszer annyiban különbözik, hogy a k_{ij} érték helyett a $\bar{k}_{ij}=k_{ij}-\alpha k_{ij}$ értéket vesszük, ahol α rendszerint a $(-0{,}1; 0{,}1)$ intervallumba esik (lásd például BURELL [11]). Ha többlépéses módszert használunk, akkor a k_{ij} fix érték helyett a $k_{ij}(x_{ij})$ értéket variáljuk. Mivel mindenegyes utazóra nem célszerű kiszámolni az útköltség értékét, feltételezzük, hogy bizonyos utazók egy csoportja választja az így variált költségű utat. Az egyszerűség kedvéért tegyük fel, hogy az egy forrásból kimenő utasok csoportja használ azonos költséget. Az így kapott költséggel számolt leggazdaságosabb utazási idejű fa útjai vagy a legrövidebb utakat, vagy az ahhoz közel eső ket szolgáltatják. A tapasztalatok szerint az egylépéses módszer és az így számolt eljárás futási ideje kevéssé tér el egymástól. Az α értékről ugyanis feltesszük, hogy csak kevés számú diszkrét értéket kaphat, amit a futás előtt generálhatunk. Az így készített egylépéses eljárás eredménye a tényleges keresztmetszeti forgalmaktól BURELL [11] szerint csupán néhány %-kal különbözik. Hasonló ötletet használ PITFIELD [54] eljárásához, aki az élhosszt *Poisson-eloszlás* felhasználásával variálja.

Az algoritmusok egy másik csoportja az egy pontba befutó élek között bizonyos arányban osztja el a forgalmat. Az elosztás különböző megfontolások alapján a pontok kezdőponttól való tényleges hosszainak arányát veszi figyelembe. Az ilyen jel-

legű modellek lehetnek egy- és kétlépésesek attól függően, hogy a h_{ij} értékek szétosztását a nyelőpontokból visszafelé egy lépésben, vagy két lépésben végezzük.

Mindkét eljárásnál szükség van az (i, j) élhez rendelt ún. élsúly függvényre. Ennek alakjára különböző szerzők mást és mást ajánlanak. A függvények közül a legegyszerűbbeket és a leggyakrabban használatosakat közöljük. Tekintsük a z pontot és jelölje A_z a z pontba bemenő élek halmazát, R_z a pontból kimenő élek halmazát. Az $i \in A_z$ ponthoz DIAL [15] a következő w_i élsúlyt rendeli hozzá:

$$w_{iz} = \begin{cases} e^{-ab_{iz}}, & \text{ha } p_i > p_z \\ 0, & \text{egyébként,} \end{cases}$$

ahol a a függvényhez rendelt paraméter, p_i a rögzített O forráspontból az i pontba vezető legrövidebb út hossza (vagy utazási ideje), $b_{iz}=p_i+t_{ij}-p_j$. A nagyméretű közlekedéstervezési programcsomagok egy része ezt a függvényformát használja (FHA [62], PLANPAC/BLACKPAC [55], TARPLAN [61] e. c. t).

Az élsúly csak olyan élekre nem nulla, amelyek végpontja távolabb van az O ponttól, mint a kezdőpontja. Nagy a esetén w_{iz} értéke kicsi — hacsak $b_{iz} \approx 0$, azaz az (i, z) él nincs a legrövidebb útban, vagy hosszban nem sokban tér el. Kis a érték esetén a w_{iz} értéke közel kerül egyhez, hacsak b_{iz} nem nagyon nagy (ez utóbbi akkor áll elő, ha hosszban nagyon távol esik az (O, i, z) út z pontba vezető legrövidebb úttól). A gyakorlatban $0 \leqq a \leqq 2$ érték látszik megfelelőnek (MORRIS [48], 173 o.). MURCHLAND [49] a GLTS számára a következő függvényt dolgozta ki:

$$w_{iz} = \begin{cases} 1 - db_{iz}/p_z, & \text{ha } db_{iz} < p_z \text{ és } p_i < p_z \\ 0, & \text{egyébként,} \end{cases}$$

ahol d a w_{iz} függvény paramétere.

Az a paraméternél elmondottakhoz hasonló összefüggés áll fenn b-re is MURCHLAND fenti képleténél. Jelöljük az $i \in N$ pontba befutó forgalom mennyiségét f_i-vel. Az *E4 egylépéses módszer* az alábbi lépésekből áll:

E40: Legyen $f_i=0$, ha $i \notin T$ és $f_i=h_{st}$, $t \in T$. Kezdetben rögzítsük le valamely $s \in S$ pontot és legyen $V=N-s$.

E41: Számoljuk ki s-ből az összes többi pontba a leggazdaságosabb utakat.

E42: Keressük meg az s pontból legtávolabbi $t \in V$ pontot.

E43: Osszuk el a t pontban levő f_t aktuális forgalommennyiséget a $j \in A_t$ pontok között a következőképp:

$$\bar{f}_j = f_t w_j / \sum_{k \in A_t} w_k$$

E44: Növeljük a $j \in A_t$ pontokban az aktuális forgalomértéket $\bar{f}_j$-vel, azaz $f_j := f_j + \bar{f}_j$.

E45: Töröljük t-t a V halmazból. Ha $V=\emptyset$ folytassuk E46-nál, egyébként E42-nél.

E46: Ha az E41—E45 lépéseket minden $s \in S$ pontra elvégeztük, fejezzük be az eljárást, egyébként válasszunk ki egy olyan $s \in S$ pontot, amellyel még nem foglalkoztunk és menjünk E41-re.

Az eljárás, mint a fenti algoritmusból is látszik igen egyszerű, mivel minden pontot csak egyszer kell megvizsgálni egy-egy forrás esetén. Így az E1 algoritmushoz képest további $O(m^2)$ számítási lépésre van szükségünk. Rendszerint egy a távolságból és utazási költségből összeállított átlagos költséggel számolják a legrövidebb utakat (lásd például [62]).

Célszerű az eljárást úgy módosítani, hogy

— induljunk ki a szabad utazási költségből;
— számoljuk ezzel végig egy tetszés szerinti forrásra az algoritmus E41—E45 lépését;
— az E46 lépés után végezzük el az E47 lépést, amely az aktuális élforgalomból kiszámolja az aktuális $k_{ij}(x_{ij})$ értéket;
— ezután végezzük rendre az E41—E47 lépéseket.

Ezzel a módosítással az esetlegesen teljesen irreálisan alacsony utazási költségeket kerüljük el. További lehetőség a többlépéses módszerek kombinációja az itt ismertetett módszerrel, ami ugyan növeli a számítási lépéseket, de pontosabbá teszi az eljárást.

A többlépéses módszer minden $s \in S$ forrás esetén kétszer számolja át a hálózat pontjait. Az s pontból kiinduló leggazdaságosabb kifeszítő fa meghatározása után egy „előre" vizsgálat, majd egy „hátra" vizsgálat történik.

Az „előre" vizsgálat során a fa pontjainak p_i potenciál szerinti növő sorrendjében meghatározzuk a

$$g_i = f_i \sum_{j \in A_i} p_j$$

értéket. Ezután az egylépéses módszert folytatjuk, de az f_i helyett a g_i értékkel, azaz az E43 lépés helyett az

$$\bar{f}_j = f_t g_j \Big/ \sum_{k \in A_t} g_k$$

értékkel. Az eljárást ezután elvégezzük minden forráspontra és a hozzá tartozó minimális hosszúságú kifeszítő fára. A két módszer közül az utóbbi minősíti pontosabban a szóba jövő utakat. Számítási és memóriaigénye viszont az előbbinek lényegesen kevesebb. Összefoglalva, a valószínűségi modellek egyszerűen programozhatók és viszonylag kevés számítást igényelnek. A futtatási eredmények arról tanúskodnak, hogy nem mindig teljesítik az equilibrium feltételt (BURELL [11], VLIET [64]). Emiatt olyan esetekben célszerű használni, amikor a hálózat telítettsége nem várható, és további analízist végzünk az eredménnyel.

Így célszerűen alkalmazhatók:

— egyes közúti közlekedési szituációk tanulmányozása esetén;
— éjszakai, illetve csúcsforgalmon kívüli városi forgalom analizálásához;
— olyan hálózatok esetén, amelyekben a $k_{ij}(x_{ij})$ függvény meghatározása nehéz vagy lehetetlen;
— fix útvonalú tömegközlekedési rendszerek tanulmányozása esetén (földalatti, külön sávos autóbusz, HÉV, vasút stb);
— olyan esetekben, ha az **O/D** mátrix csak egy kis részhalmazával kell dolgoznunk (taxi, teherszállítás stb).

1.4.4. *Feladat megoldása matematikai programozással*

Az 1.1 pontban felsoroltuk a matematikai programozást használó forgalomelosztási módszereket. A fejezetben egy eljárást mutatunk be, amely a FRANK—WOLFE [26] algoritmus alapötletét használja egyéni optimális feladat megoldására.

A konvex programozási feladat a következő:

$$f(\mathbf{x}) \to \min \tag{1.36}$$

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{1.37}$$

$$\mathbf{x} \geqq \mathbf{0}, \tag{1.38}$$

ahol $\mathbf{A}$ $m \times n$-es mátrix, $f(\mathbf{x})$ pedig konvex függvény. Jelöljük az (1.36), (1.37) feltételnek eleget tevő megoldások halmazát M-mel (ez az ún. lehetséges megoldások halmaza).

Tegyük fel, hogy

I. Az f függvény folytonosan differenciálható;

II. Bármely $\mathbf{a} \in M$ esetén a $\langle \nabla f(\mathbf{a}), \mathbf{x} \rangle$ lineáris függvény alulról korlátos az M halmazon.

A *Frank—Wolfe-algoritmus* az alábbi lépésekből áll: Konstruáljuk meg az $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k, \ldots$ megoldások halmazát úgy, hogy

$$\lim_{k \to \infty} f(\mathbf{x}_k) = \min_{\mathbf{x} \in M} f(\mathbf{x}).$$

Legyen $\mathbf{x}_0$ egy megengedett kiinduló megoldás. Ezt megkaphatjuk például (1.36), (1.37) megoldásával. Az $\mathbf{x}_k \in M$-ből a fenti feltételeknek eleget tevő $\mathbf{x}_{k+1}$ megoldást a következő eljárással kapjuk:
megoldjuk a lineáris feladatot a következő célfüggvény minimalizálásával:

$$\langle f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle \to \min, \quad \mathbf{x} \in M.$$

Ez a célfüggvény minimalizálás equivalens a következő minimalizálással:

$$\langle \nabla f(\mathbf{x}_k), \mathbf{x} \rangle \to \min, \quad \mathbf{x} \in M.$$

és II. miatt ennek van $\bar{\mathbf{x}}_k$ optimális megoldása.

Két eset lehet a

$$g(\mathbf{x}) = \langle \nabla f(\mathbf{x}_k), \mathbf{x}_n - \mathbf{x}_k \rangle$$

alak vizsgálatával kapcsolatban:

a) ha $g(\bar{\mathbf{x}}_k) \geqq 0$, akkor minden $\mathbf{x} \in M$ esetén $g(\mathbf{x}) \geqq 0$, ugyanis $f(\mathbf{x}) - f(\mathbf{x}_k) \geqq g(\mathbf{x})$, mivel $\partial f(\mathbf{x}_k, \mathbf{x} - \mathbf{x}_k) = g(\mathbf{x})$;
b) ha $g(\bar{\mathbf{x}}_k) < 0$, akkor az $\mathbf{x}_k$ pontból az $\bar{\mathbf{x}}_k$ felé $f(\mathbf{x})$ értéke csökken.

Válasszunk ki egy olyan $\mathbf{x} \in [\mathbf{x}_k, \bar{\mathbf{x}}_k]$ pontot, amelyre $f(\mathbf{x})$ felveszi a minimumát. Azaz megoldjuk az alábbi feladatot:

$$\min \{ f(\mathbf{x}_k + \lambda(\bar{\mathbf{x}}_k - \mathbf{x}_k)) | 0 \leqq \lambda \leqq 1 \}.$$

Legyen a megoldás λ_k, és $\mathbf{x}_{k+1}$-et válasszuk a következőképp:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k(\bar{\mathbf{x}}_k - \mathbf{x}_k).$$

Nyilvánvaló, hogy

$$f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k).$$

Meg lehet mutatni, hogy a fenti eljárás konvergens. Ezt az eljárást fogjuk alkalmazni a konvex célfüggvénnyel rendelkező egyéni optimális feladat megoldására.

A célfüggvény eleget tesz az I. feltételnek, mivel a parciális deriváltak léteznek és megfelelő $k_{ij}(x_{ij})$ függvényt választva folytonosak is. A II. feltétel is teljesül, mivel a megoldáshalmaz konvexp olitóp, a $k_{ij}(x_{ij})$ függvény pedig konvex pozitív függvény a megoldást adó legrövidebb utak költsége pedig véges.

A q-adik lépésben itt is egy lineáris programozási feladatot kell megoldani. A célfüggvény

$$c_{ij}(x_{ij}) = \int_0^{x_{ij}} k_{ij}(y)\,d_y \tag{1.39}$$

függvény egyszerű deriváltjaiból adódik, ennek deriváltja pedig éppen a $k_{ij}(x_{ij})$ függvény. A megoldás a mátrix speciális struktúrája miatt egy legrövidebb út feladat megoldásával ekvivalens, mégpedig ezek sorozatának felhasználásával egy mindent — vagy — semmit eljárást kell megoldani $k_{ij}(x_{ij})$ függvénnyel.

A lépéshosszt az eredeti eljáráshoz hasonlóan egy egydimenziós minimalizálási feladat eredményeképp kaphatjuk meg. Erre külön nem térünk ki, egy sor ilyen eljárást ír le LUENBERG [42] (7. fejezet 133—167 o.). Összefoglalva az itt elmondottakat, az alábbi E5 algoritmust kapjuk:

E50: Határozzunk meg egy $X=(x_{ij})$ lehetséges megoldást ($k_{ij}(0)$ költséggel).

E51: Határozzunk meg mindent vagy semmit algoritmussal az $\mathbf{x}=(x_{ij})$ megoldáshoz tartozó $k_{ij}(x_{ij})$ költséggel egy $\bar{\mathbf{x}}=\bar{\mathbf{x}}_{ij}$ megoldást.

E52: Ha $\left|\sum_{ij}(x_{ij}-\bar{x}_{ij})\,k_{ij}(x_{ij})\right|<\varepsilon$ készen vagyunk, egyébként megyünk tovább.

E53: Határozzuk meg a λ^* lépésközt egydimenziós optimalizálással, amely minimalizálja a következő összeget:

$$\sum_{i,j} c_{ij}(x_{ij})(x_{ij}+\lambda(\bar{x}_{ij}-x_{ij})) \to \min,$$

$$0 \leqq \lambda \leqq 1,$$

ahol c_{ij}-t (1.39)-ben adtuk meg.

E54: Módosítsuk az x_{ij} értékét:

$$x_{ij} = x_{ij} + \lambda^*(\bar{x}_{ij} - x_{ij})$$

és folytassuk E51-nél.

A feladat egy kvadratikus költségfüggvényes változatát írja le dolgozatában KAS—MAYER [35].

2. Leggazdaságosabb útvonalak meghatározása

Mint az előző fejezetben láttuk, a forgalomelosztási probléma központi kérdése a leggazdaságosabb (legkisebb költségű, legrövidebb utazási idejű, a továbbiakban minimális) útvonal meghatározása. Ugyanis az eljárásnak gyorsasága a nagyszámú útvonal meghatározásának gyorsaságától függ. Hasonlóan felhasználást nyer a problémakör egyéb tervezési fázisokban is. Így használjuk a forgalomgenerálás, elosztás megoldásánál is. Az optimális úthálózat meghatározása is igényli a hatékony legrövidebb út meghatározási eljárásokat.

Az 1. fejezetben bevezetett jelöléseknek megfelelően az s pontból a t pontba vezető utat P_{st}-vel jelöljük, és a $P_{st}=(s=x_0, x_1, \ldots, x_{v+1}=t)$ útvonal hosszán az

$$l(P_{st}) = \sum_{i=1}^{v} t_{x_i x_{i+1}} \tag{2.1}$$

értéket értjük. Legrövidebb út a $\bar{P}$, amelyre $l(\bar{P}_{st})=\min_{P_{st}} (P_{st})$.

A P út hosszával kapcsolatban a következő feladatokat fogalmazhatjuk meg:

a) Határozzuk meg az $s, t \in N$ rögzített pontok között azt a P_{st} utat, amelyre (2.1) minimális.
b) Határozzuk meg a rögzített $s \in N$ pontból a hálózat összes $i \in N$ pontjába a P_{si} legrövidebb utakat.
c) Keressük meg a hálózat minden $i, j \in N$ pontpárja között a P_{ij} legrövidebb hosszúságú utakat.
d) Határozzuk meg a fenti a—c feladatok valamelyik változatára az első k számú legrövidebb utat.

A fenti feladatok mellett továbbiak is megfogalmazhatók, mint rögzített k ponton átmenő, rögzített éleken átmenő, legfeljebb l számú rögzített pontot (élt) tartalmazó útvonalakkal kapcsolatos problémák. Ezek tárgyalásától eltekintünk, mivel a dolgozatban közölt témákban nem használjuk őket.

Megjegyezzük, hogy az a) és b) feladat csak az elérendő cél szempontjából különböző, a megoldást azonos eljárásokkal határozhatjuk meg. A c) feladat az a) vagy a b) feladatot megoldó algoritmusok többszöri alkalmazásával is megoldható.

Az irodalom a feladat fontosságának megfelelően közel 400 cikket tartalmaz a problémakörrel kapcsolatban. A különféle módszerek közül mi csak az alapeljárásokat és a közlekedési hálózatok céljaira kifejlesztett futási és tárolási idő szempontjából leggazdaságosabb algoritmusunkat ismertetjük. További részletes összefoglalót tartalmaznak POLLACK—WIEBENSON [56], DREIFUS [18], EDMONDS—KARP [19], GOLDEN [28], VLIET [65], és BAKÓ [3, 4] dolgozatai.

2.1. *Faépítő módszerek*

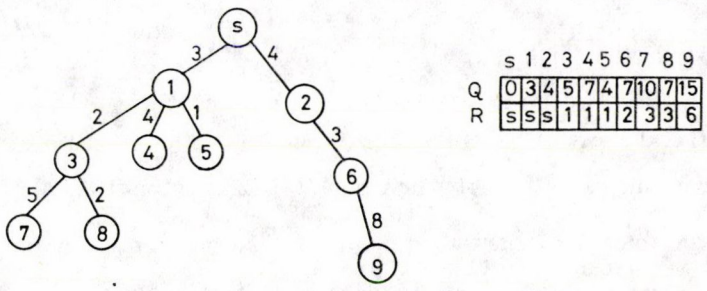
A fejezetben tárgyalt módszerek mindegyike egy minimális összhosszúságú fát határoz meg. Ennek csúcspontja — egy előre rögzített s pont — tartalmaz minden $i \in N$ pontot (feltételezve, hogy a hálózat összefüggő), és az x_s ponttól bármelyik $i \in N$ pontig a legrövidebb P_{si} útvonalból áll. Ezt a fát szokás az s gyökérpontú minimális hosszúságú kifeszítő fának (vagy röviden minimális fának) nevezni. Ezen mód-

szerek az előző pontban megfogalmazott a) és b) feladat megoldását szolgáltatják. Ha minden $s \in N$ esetén végrehajtjuk az eljárást, akkor a c) feladat megoldását is megkapjuk.

Egy s pontban kezdődő minimális fa célszerűen két vektorral adható meg. Az egyik a **Q** potenciálvektor, a másik az **R** címkevektor. A **Q** vektor q_i eleme adja meg az $l(P_{si})$ távolságot, az **R** vektor r_i eleme azt a pontot adja meg, amelyik a P_{si} úton az i pontot megelőzi. Így az útvonalat magát az **R** vektor segítségével az i ponttól visszafelé rakhatjuk össze: a $P_{si}=(x_1, x_2, \ldots, x_v)$ esetén

$$i = x_v, \quad r_i = x_{v-1}, \quad r_{x_{v-1}} = x_{v-2}, \ldots, r_{x_2} = x_1 = s.$$

A **Q** és **R** vektorok felépítését a 3. ábrán levő fa esetén mutatjuk be.



3. ábra
Potenciál és címkevektor

A fa élein az utazási költséget tüntetjük fel. A faépítő algoritmusok a **Q** és **R** vektorok kitöltését végzik. Három alapmódszert különböztetünk meg. Az egyik ún. potenciál módszert FORD [25] és MINTY [45] készítette. Ennek egy módosított, számítógépre alkalmasabb változatát írja le DIJKSTRA [17]. Egy, az előzőtől elvileg különböző potenciálmódszert hozott létre D'ESOPO (lásd POLLACK—WIEBENSON [56]). BELLMAN [9] dinamikus módszere szintén alkalmas a feladat megoldására, de számításigényessége miatt nem foglalkozunk vele (BAKÓ [3]).

Ford—Minty-algoritmushoz N pontjait két csoportba osztjuk: $N=S \cup T, S \cap T = \emptyset$. Az $x \in S$ pontjainak q_x potenciálja ismert. Ez a legrövidebb út hossza a rögzített $s \in S$ pontból az x pontig. Minden lépésben egy ponttal bővítjük az S halmazt, így n lépésben meghatározzuk a minimális kifeszítő fát, amivel a b) feladatot megoldottuk. Az a) feladatot akkor oldottuk meg, ha a rögzített másik végpont is bekerül az S halmazba.

Az L1 algoritmus formális leírása a következő:

L10: $r_s=s$, $q_s=0$, $q_i=\infty$, $r_i=0$, ha $i \neq s$, $S=\{s\}$, $T=N-s$

L11: Legyen

$$q_v + t_{vw} = \min_{\substack{i \in S \\ j \in T}} (q_i + t_{ij})$$

L12: $q_w=q_v+t_{vw}$, $r_w=v$, $S=S+v$, $T=T-v$

L13: Ha $T=\emptyset$, akkor STOP, egyébként folytassuk L11-nél.

Az algoritmus során az L11 lépésben legfeljebb $N/2$ összeadást és összehasonlítást kell végezni. Az L12 lépésben összesen 2 értékadás és az S halmaz növelése (T csökkentése) szerepel.

Ez utóbbit nem kell külön regisztrálnunk, mivel azon x pontok tartoznak S-be, amelyekre $q_x<\infty$, és T-be, amelyekre $q_x=\infty$. Az összes összeadások és összehasonlítások száma $n^3/3$. Az algoritmus jelentősen gyorsítható. Egy ilyen módosított változatot készített DIJKSTRA [17], amit az alábbiakban foglalunk össze.

A módszer lényege az, hogy bizonyos S halmazba bekerülendő pontok az L11 lépésben való összeadáskor értéket kapnak, amit a továbbiakban fel fogunk használni. Ezeket az értékeket ideiglenes potenciáloknak fogjuk nevezni. Az N halmazt most három diszjunkt halmazra bontjuk $N=S\cup A\cup T$, hasonlóképp az E élhalmazt is $E=X\cup Y\cup Z$.

A halmazok definíciója a következő:

S: a már végleges potenciállal rendelkező pontok halmaza,

$$A=\{y|x\in S,\ y\notin S,\ (x,y)\in E\},$$

$$T=N\cap(S\cup A),$$

$$X=\{(x,y)|x,y\in S,\ (x,y)\in E\},$$

$$Y=\{(x,y)|x\in S,\ y\in A,\ (x,y)\in E \text{ és minden } y\text{-hoz csak egy él tartozik}\},$$

$$Z=E\cap(X\cup Y).$$

Legyen az utoljára S-be került pont v, és jelölje $H=\{v,y\}\ y\in A$.
Az L2 algoritmus formális leírása a következő:

L21: Ha $y\in T$, $(v,y)\in H$, akkor $A=A+y$, $Y=Y+(v,y)$, $q_y=q_v+t_{vy}$, máskülönben L22-nél folytassuk.

L22: Minden $y\in A$-ra végezzük el a következőt:

Ha $q_v+t_{vy}<q_y$, $y\in A$, akkor

$Y=Y+(v,y)$, $Y=Y-(x,y)$,

és (x,y) törölhető az élek halmazából.

L23: Legyen $q_z=\min\limits_{i\in A} q_i$, $r_z=V$

$$S=S+z,\quad A=A-z,\quad X=X+(v,z),$$

$$Y=Y-(v,z)$$

L24: Ha A üres (így T is üres), készen vagyunk, egyébként folytatjuk L20-nál.

Mint látjuk, a potenciálvektor meghatározása nehézkesebb, viszont az algoritmussal $n^2/2$-re csökkentettük az összeadások és $2n^2$-re az összehasonlítások számát.

A harmadik potenciálmódszer alapötletében különbözik az előző kettőtől. Az eddigi módszereknél ugyanis minden S halmazbeli pont végleges potenciállal rendelkezett, a *D'Esopo-módszernél* viszont minden pont potenciálja ideiglenes egé-

szen addig, ameddig az eljárást be nem fejeztük. Az L3 algoritmust az alábbiakban adjuk meg:

L30: Legyen $S=N$, $q_s=0$, $q_i=\infty$, ha $i\in N$, $r_s=1$, $r_i=0$.

L31: Válasszunk egy $x\in S$ pontot, amelyre $q_x\neq\infty$.

L32: Az összes $(x, y)\in E$ esetén végezzük el az alábbi műveleteket:

Ha $q_y<q_x+t_{xy}$, akkor $q_y=q_x+t_{xy}$, $r_y=x$ és ha $y\notin S$, $S=S+y$

L33: $S=S-x$; ha S üres, menjünk L31-re, egyébként készen vagyunk.

Ez utóbbi eljárás annak ellenére, hogy az összes S-beli elemet végigveszi, bizonyos esetekben gyorsabb lehet, mint az előzők. Nevezetesen, akkor, ha az élek azonos hosszúságúak. Ez fordul elő például olyan feladatoknál, ahol az úthossz helyett az élek számának minimalizálására törekszünk, azaz minden élhossz egységnyi.

A tárolási igényre, gyorsaságra és programozhatóságra a fejezet végén visszatérünk. A fenti algoritmusok helyességének bizonyítására sem térünk ki külön, mivel azt a BAKÓ [3] részletesen tárgyalja.

2.2. *Mátrix eljárások*

A mátrix eljárások a $\mathbf{T}=(t_{ij})$ költségmátrixon végeznek egymás utáni operációkat. Az eljárások eredményeképp megkapjuk a $\mathbf{Q}^{(p)}=(q_{ij}^{(p)})$ mátrixot, amelynek $q_{ij}^{(p)}$ eleme tartalmazza a P_{ij} minimális út hosszát. Egy további $\mathbf{R}=(r_{ij})$ $n\times n$-es mátrix megfelelően vezetve a címkemátrix szerepét fogja játszani. Az útvonal visszakeresése viszont nem egyezik meg a címkevektornál elmondottakkal, így azt külön ismertetjük. Két alapmódszert tárgyalunk. A többi ennek többé-kevésbé módosított változata. Az eljárásokkal részletesen foglalkozik BAKÓ [4] dolgozata. Az egyik könnyen programozható, elegáns módszert WARSHALL [69], programját pedig FLOYD [24] közölte.

Jelöljük az (i, j) pontpár közötti és közbülső pontként legfeljebb az $1, 2, \ldots, k$ pontok valamelyikét tartalmazó legrövidebb utat $P_{ij}^{(k)}$-val, az összes (i, j) pár esetén az ezen utakhoz tartozó potenciál mátrixot $\mathbf{Q}^{(k)}$-val. Az eljárás során egymás utáni lépésekben meghatározzuk a $\mathbf{Q}^{(1)}, \mathbf{Q}^{(2)}, \ldots, \mathbf{Q}^{(n)}$ mátrixokat:

Az L4 algoritmus lépései a következők:

L40: Legyen $k=1$, $\mathbf{Q}^{(0)}=(t_{ij})$, $r_{ij}^{(0)}=j$, $i=1, 2, \ldots, n$.

L41: Határozzuk meg a $\mathbf{Q}^{(k)}=(q_{ij}^{(k)})$ és az $\mathbf{R}^{(k)}=(r_{ij}^{(k)})$, elemeket a következőképp:

$$q_{ij}^{(k)}=\begin{cases} q_{ik}^{(k-1)}+q_{kj}^{(k-1)}, & \text{ha } q_{kj}^{(k-1)}>q_{ik}^{(k-1)}+q_{kj}^{(k-1)} \\ q_{ij}^{(k-1)}, & \text{egyébként.}\end{cases}$$

$$r_{ij}^{(k)}=\begin{cases} r_{ik}^{(k-1)}, & \text{ha } q_{ij}^{(k-1)}>q_{ik}^{(k-1)}+q_{kj}^{(k-1)} \\ r_{ij}^{(k-1)}, & \text{egyébként.}\end{cases}$$

L42: Ha $k=n$, akkor készen vagyunk, egyébként $k=k+1$ és menjünk L41-re.

Könnyen belátható, hogy az összes összeadások és összehasonlítások száma n^3, mivel egy lépésben n^2 műveletet végzünk és $k=1, 2, \ldots, n$-re hajtjuk végre az L41 lépést. Ennek ellenére a módszer kevésbé alkalmazható nagy méretek esetén a tárolásigényesség miatt. Nagy méretek esetén HU [32] a módszer igen gyors dekompozíciós változatát javasolja.

Tapasztalataink szerint a hálózat dekompozícióját nehéz megadni, és a programozása is igen komplikált (lásd KIRÁLY—BAKÓ [37].

A másik mátrix módszer dinamikus programozási elvvel dolgozik (BELLMAN [9], SHIMBEL [59], KALABA [34]). A dinamikus elvet abban az értelemben használjuk, hogy egy út akkor és csakis akkor minimális, ha bármely része is az. Az előző módszerrel szemben itt rendre a legfeljebb 2, legfeljebb 3, ..., legfeljebb $(n-1)$ élből álló legrövidebb utakat fogjuk meghatározni.

Általában $\mathbf{Q}^{(k)}$ fogja tartalmazni az összes pontpár között a legrövidebb utak hosszát, amelyek legfeljebb k élből állnak.

Az L5 algoritmus lépései a következők:

L50: Legyen $\mathbf{Q}^{(1)}=(t_{ij})$ és $k=1$.

L51: Számoljuk ki a $\mathbf{Q}^{(k+1)}$ mátrix elemet a következőképp

$$q_{ij}^{(k+1)} = \min_{1 \leq f \leq n} (q_{if}^{(k)} + q_{fj}^{(k)})$$

L52: Ha $k=n-1$ készen vagyunk, egyébként $k=k+1$ és menjünk L51-re.

A módszer egyszerű, de viszonylag sok számolást igényel, mivel az összeadások és összehasonlítások száma lépésenként n^3, így összesen n^4 lépést kell végezni. Meg lehet mutatni, hogy

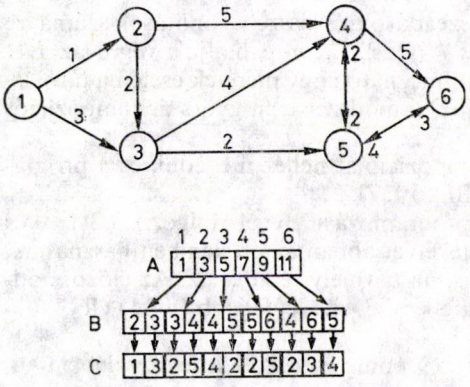
$$q_{ij}^{(l+k)} = \min_{1 \leq f \leq n} (q_{if}^{(l)} + q_{fj}^{(k)})$$

A leggyorsabb 2 hatványai szerint haladni és így elegendő a $\mathbf{Q}^{(2)}, \mathbf{Q}^{(4)}, \ldots$ mátrixokat kiszámolni. Ez az eljárás $\log^2(n-1)n^3$ számítási lépést igényel. A mátrix algoritmusok használhatóságáról a fejezet végén bővebben beszélünk.

2.3. *Új algoritmusok nagyméretű közlekedési hálózatokra*

Az eddig ismertetett módszerek hibája az, hogy nem használják ki a közlekedési hálózatok speciális szerkezetét. Ezen hálózatokra az jellemző, hogy a pontok növekedésével egyre ritkábbak lesznek, mivel egy csomópontból átlagosan 3—4 él megy ki.

A fejezetben 3 olyan eljárást ismertetünk, amelyek nagy méretek esetén a legjobb eredményt nyújtják. Ezekből az első kettőt a szerző dolgozta ki (L6 és L7), a harmadik a DIAL [15] ötletének továbbfejlesztéséből adódott (L8). Az algoritmusok ismertetése előtt megadjuk a számítógépes szempontból legjobb hálózattárolási lehetőséget. Az incidencia mátrix n^2, a honnan-hova mátrix $3m$ tárolóhelyet igényel. Ezekkel az ábrázolásokkal kapcsolatban a probléma tárolandó elemek nagy száma incidencia mátrix), illetve az egy pontból kiinduló élek megkeresésének nehézkessége (honnan-hova mátrix). Ezen problémák kikerülésére készítettünk egy tárolási eljárást. Ennél a hálózat struktúrájának megadásához és a távolságadatokhoz összesen 3 vektor szükséges. Jelöljük ezeket **A**, **B**, **C**-vel. Az **A** egy mutató vektor, melynek a_i



4. ábra
Hálózat megadása vektorokkal

eleme azt mutatja, hogy az i-edik pontból kimenő élek végpontjai, illetve ezen élek távolságára vonatkozó adatok hol kezdődnek a **B** végpont, illetve a **C** távolság vektorban. Így **A** hossza n, **B** és **C** hossza m, azaz a hálózat tárolásához $n+2m$ hosszúságú tárolóhelyre van szükség. Egyéb sűrített tárolási módszereket is használhatunk (l. BAKÓ [7]), de céljainknak ez a legmegfelelőbb.

A hálózat ilyen megadását mutatjuk be a 4. ábrán.

Az ismertetendő mindhárom eljárás potenciálmódszer.

Az első eljáráshoz rendezzük le az egy pontból kifutó éleket nagyság szerint a **C** vektorban. Mivel csak néhány él megy ki egy pontból, így a **C** (és ezzel együtt a **B**) vektor elemeinek rendezése pontonként csak néhány összehasonlítást igényel. Ezután rendeljünk hozzá a hálózat minden éléhez egy további d_i elemet, amely az i pontból kimenő és még minimális fában nem levő élek legrövidebbikére mutat. Így lépésenként minden $i \in S$-re csak egy összeadást kell végeznünk. Az S pontok halmazát **Q** potenciálvektor végignézésével kapjuk meg, mégpedig $i \in S$, ha $q_i \neq \infty$. Ehelyett egy L listán tartjuk az S olyan elemeit, amelyek egyáltalán szóba jöhetnek az L11 lépésben, amely feleslegessé teszi a **Q** vektor ismételt végignézését. Az L lista aktuális hosszát egy további f változóban tartjuk.

Így a módosított L6 algoritmus a következő lépésekből áll:

L60: $l_1=s$, $l_i=0$, $f=1$, $q_s=0$, $q_i=\infty$, $r_s=s$, $r_i=0$, $\mathbf{D}=\mathbf{A}$, $k=1$.

L61: Legyen $y=q_v+c_w=\min\limits_{\substack{i=l_i \\ l_i\neq 0}} (q_i+c_{d_i})$.

L62: $d_v=d_v+1$, $q_w=y$, $r_{b_w}=v$, $k=k+1$;
ha $d_v=a_{v+1}$, akkor $l_v=b_w$, egyébként $l_{l+1}=b_w$ és $f=f+1$.

L63: Ha $b_w=t$ (a feladat), illetve $k \geqq n$ (b feladat) készen vagyunk, egyébként menjünk L61-re.

A fenti eljárás előnye az L1 módszerhez képest, hogy

— nem számoljuk ki az összes szóbajöhető élre az új lehetséges potenciált, csak minden L-beli pontból egy élre;

— az L listán csak az aktuális S-beli pontok szerepelnek.

Az algoritmusban összesen k összeadást és összehasonlítást kell lépésenként végezni, ahol k az $L \subseteq S$ halmaz számossága. A módszert továbbfejlesztjük úgy, hogy minden lépésben egy vagy két összeadást, és az ezen értékek egy már rendezett vektorba való elhelyezését kell elvégezni. Így az összes összeadások száma n, az összehasonlítások száma $n \cdot k/2$ és nk közé esik.

Legyen E_{ST} a lehetséges élek halmaza, azaz

$$E_{ST} = \{(x, y) | x \in S, y \in T, (x, y) \in E, t_{xy} = \min_{z \in T} t_{xz}\}.$$

Az E_{ST} halmazbeli élek kezdőpontjait a **H1**, végpontjait a **H2** vektorok tartalmazzák. Az **R** vektorban az ezen pontokhoz tartozó ideiglenes potenciálok szerepelnek (61-ben számolt értékek!). Az **R** vektor nagysága szerint növő sorrendben le van rendezve.

Az algoritmus alapgondolata az, hogy az E_{ST}-beli élekre a potenciálokat csak a bekerüléskor határozzuk meg, és így elkerüljük a lépésenkénti nagyszámú összeadást. Az E_{ST} halmaz lépésenként legfeljebb csak 2 éllel bővül, az egyik az éppen a minimális fába került és kezdő, a másik a végpontjából indul ki.

Ezekre kiszámítva az ideiglenes potenciálokat, és összehasonlítva a már kiszámolt értékekkel azonnal megkapjuk a fába bevonandó élt.

Az L7 algoritmus formális leírása a következő:

L70: $S=s$, $T=N-S$, $p=B(A(s))$, $E_{ST}=(s, p)$,
$L=s$, $\mathbf{H1}(1)=s$, $\mathbf{H2}(1)=p$, $D=\mathbf{A}$, $\mathbf{R}(1)=t_{sp}$.
Tételezzük fel, hogy az utolsó lépésben a (v, w) éllel bővítettük a minimális fát.

L71: $E_{ST}=E_{ST}-\{(v, w)\}+\{(w, p), (v, q)\}$,
ahol $t_{wp}=\min_{j \in T} t_{wj}$, $t_{vq}=\min_{j \in T} t_{vj}$.

L72: Számoljuk ki a p és q pontokhoz tartozó $\bar{p}$, $\bar{q}$ ideiglenes potenciál értékeket.

L73: Rendezzük be az **R** vektorba a $\bar{p}$, $\bar{q}$ értékeket, és helyezzük el a v, w kezdő, illetve p, q végpontokat a **H1**, illetve **H2** vektorba a megfelelő helyre.

L74: A (**H1**(1), **H2**(1)) él kerül a minimális fába, így elhagyjuk a **H1**, **H2** és **R** első elemét, **H2**(1)-et az S halmazból a T halmazba tesszük.

L75: Ha T üres készen vagyunk, egyébként L71-nél folytatjuk az eljárást.

Az L7 algoritmus előnye az L6-tal (és a korábbiakkal) szemben nyilvánvaló, a különböző vektorok kezelése azonban bonyolultabbá teszi a program megírását. Helyigénye $4n+2m+3k$, ahol k az E_{st} halmaz maximális számossága ($k<n$).

Az L6—L7 algoritmusokhoz hasonlóan az L8 is potenciál módszer. Az ideiglenes potenciálokat minden élre csak egyszer számoljuk ki az L7-hez hasonlóan. Ezek rendezése viszont elmarad, mert egy **K** ideiglenes potenciálokat tartalmazó vektorba a nagyságnak megfelelő helyre kerülnek automatikusan.

Ehhez azt a nyilvánvaló tényt használjuk ki, hogy a faépítés során egy-egy lépésben szóbajöhető élek legfeljebb $t=\max_{i,j} t_{ij}+1$-gyel térhetnek el egymástól. Így az ideiglenes potenciált mod t-re kiszámolva elhelyezzük a **K** vektorba. Az azonos ideiglenes potenciálok helyét egy **L** láncvektorral kapcsoljuk össze. Egy változóban külön számoljuk, hogy mennyi az aktuális „szorzószám" a valódi potenciálok megkapásához. A módszernél is felhasználjuk az L7-es eljárásban alkalmazott mutatóvektor aktualizálást, így nem kell megkeresni a még fában nem levő pontokat.

Az L8 eljárás az alábbi lépésekből áll:

L80: $l_i=0$, $k_i=0$, $S=s$, $T=N-S$, $E_{ST}=(s, j)$, $r_i=0$, $q_i=\infty$, $r_1=s$, $q_1=0$,

L81: Legyen

$$d_j = q_i + t_{ij}, \quad i \in S, \; j \in T.$$

L82: Ha $q_j > d_j$, akkor

— ha $q_j = \infty$, akkor $q_j = d_j$ és $\bar{d}_j = d_j \pmod t$ és $k_{\bar{d}_j} = j$;
— ha $q_j \neq \infty$, akkor az **L** láncból kivesszük a q_j-hez tartozó végpontot és $k_{\bar{d}_j}$-hoz tartozó **L** láncba betesszük j-t, egyébként L83-nál folytatjuk.

L83: Ha L81—L82-t minden $(i,j) \in E_{ST}$ élre elvégeztük, megyünk L84-re, egyébként L81-re.

L84: Megkeressük a következő legkisebb potenciált **K**-ban, és az ahhoz tartozó végpontot S-be tesszük (T-ből kivesszük).

L85: Ha minden $i \in N$ esetén $i \in S$, készen vagyunk, egyébként L81-nél folytatjuk.

2.4 *Módszerek összehasonlító vizsgálata*

A fejezetben összefoglaljuk az előző pontokban ismertetett eljárásokkal kapcsolatos tapasztalatainkat. Az elméletileg becsült lépések számát és a memóriaigényt az 1. táblázat mutatja.

1. TÁBLÁZAT

Módszerek erőforrás igényei

	Lépések száma		Adatok	Munka-ter.	Pontok sz.	Élek sz.	$p \geqq n$	$q \geqq m$	Utasítások száma	Memória igény
	összeadás	hasonlítás								
Warshall	n^3	n^3	n_2	n_2	100	N	p^2-100_2	—	19	850
Dinamikus	n^4	n^4	n_2	n_2	100	N	p^2-100_2	—	19	850
Ford	$n^4/2$	$n^4/2$	n_2	$2n$	100	N	$4(p-n)$	$2(q-m)$	31	1032
D'Esopo	N	N	$n_2 2m$	$2n$	100	N	$4(p-n)$	$2(q-m)$	28	1000
L6	$N^3/2$	$n^3/2$	$n+2m$	$3n$	5000	15 000	$4(p-n)$	$2(q-n)$	31	1200
L7	$v \cdot w$	$v \cdot w$	$n+2m$	$6n$	3500	14 000	$6(q-n)$	$2(q-m)$	97	2560
L8	N	N	$n+2m$	$3n+1$	5000	15 000	$4(p-n)$	$2(q-m)$	59	1700

$v = n-1$, $w = m-n$

A módszereknél a hálózat minden pontjából kimenő minimális hosszúságú kifeszítő fára vonatkozó adatokat adjuk meg. A hálózat ábrázolásához szükséges memóriaigényt az L6—L8 módszereknél a 4. ábrán megadott tárolást alkalmazva számoltuk ki. A 3. oszlop a hálózat ábrázolásához szükséges adatterületet tartalmazza (egy egység egy egész szám ábrázolásához szükséges memóriaterület). Mint látható, a hálózat tárolása szempontjából legelőnyösebb az L6—L8 módszer. Munkaterület szempontjából is ezek a legjobb eljárások. Az L8 módszernél l a max $t_{ij}+1$ értéket jelöli. A következő két oszlop az elkészült programok maximális pontjainak (éleinek) a számát adják. Azokon a helyeken, ahol nincs megkötés az élek számára, oda N betűt írtunk a 6. oszlopba. A következő oszlop a pontok számának n-ről p-re való növeléséhez szükséges memóriaigényt adja meg. A 8. oszlop ugyanazt jelzi az élek számának m-ről

q-ra történő bővítése esetén. A két utolsó oszlop a program bonyolultságát (utasítások száma), illetve memóriaigényét mutatja.

A programokat R20-as számítógépre és FORTRAN nyelven készítettük el. A leghasználhatóbb módszerek gépi kódú változata is elkészült (*D'Esopo*, L7, L8).

A számítógéppel történő összehasonlításhoz két hálózatgeneráló eljárást készítettünk el. Az egyik eljárással tetszésszerinti pontszámú és sűrűségű hálózatot lehet előállítani mátrix formában vagy a 4. ábrán megadott módon. A program bemenő paraméterei a pontok száma, a sűrűség %-ban és az utazási értékek maximuma. A távolságértékeket könyvtári véletlenszám generátorral (RANDU) állítjuk elő (1 és a megadott maximum értékek között).

A pontok számát tekintve 10, 20, 40, 60, 80 és 100 pontos hálózatot generáltunk 5, 10, 30, 50, 80 és 100%-os sűrűséggel. Kevés pontszámú ritka hálózat előállítása nem célszerű, így a 10, 20 és 40 pontos hálózat 5%-os sűrűségű és a 10 és 20 pontos hálózat 10%-os sűrűségű változatával nem foglalkoztunk.

A módszerek közül a *Warshall*, a *Dinamikus*, a *Ford*, a *D'Esopo*, az L6, L7, és L8 módszert programoztuk be. A részletes futtatási eredményeket a Mellékletekben adjuk meg.

A másik generáló programmal négyszög hálózatokat lehet generálni. Egy $m \times n$-es méretű négyszög hálózat egy m sorból és n sorból álló rács hálózatnak felel meg, amely legjobban megközelíti a valós úthálózatot. Ezt a soronként és oszloponként ekvidisztáns távolságú hálózatot torzítjuk úgy, hogy minden élre kiszámolunk egy 1 és 10 közötti értéket a RANDU rutin felhasználásával. A négyszög hálózat generáló program bemenő paraméterei az x és y (sor és oszlopszám), a generált hálózat vagy mátrix vagy a 4. ábrán megadott alakban kapható meg.

A *Warshall, Dinamikus, Ford* és *D'Esopo módszert* két változatban készítettük el. Az egyik csak az útvonalak hosszát, a másik az útvonalat magát is megadja. Az L6—L8 módszereknél a címke is szerepet játszik az eljárásban, ezért ezekre külön nem készült ilyen változat.

Az elkészült programokat különböző hálózatokon teszteltük. A kitűzött cél a módszerek hatékonyságának összehasonlítása a következő szempontokból:

— pontok száma,
— élek száma,
— csak utazási idők, és útvonalak és utazási idők,
— általános és négyszög hálózatok.

Pontok száma és futási idő

A futási idő növekedésének a pontos számától való függését tömör és ritka hálózatokon egyaránt megvizsgáltuk. Csak a teljes és a 30%-os sűrűségű hálózatok esetén vizsgáltuk meg, hogy milyen függvénnyel lehet legjobban közelíteni az összefüggést. A tiszta gépidőket teljes hálózatra vonatkoztatva a 2. táblázat tartalmazza. Ebben a különböző méretű általános és minden élt tartalmazó tömör hálózatok esetén kapott futtatási időket adjuk meg másodpercben.

A módszereket a fenti hat generált tömör hálózaton teszteltük. Mint várható, a leggyorsabb ilyen hálózatokra a *Warshall módszere*. Ez minden hálózaton jobb eredményt adott mint az egyéb módszerek.

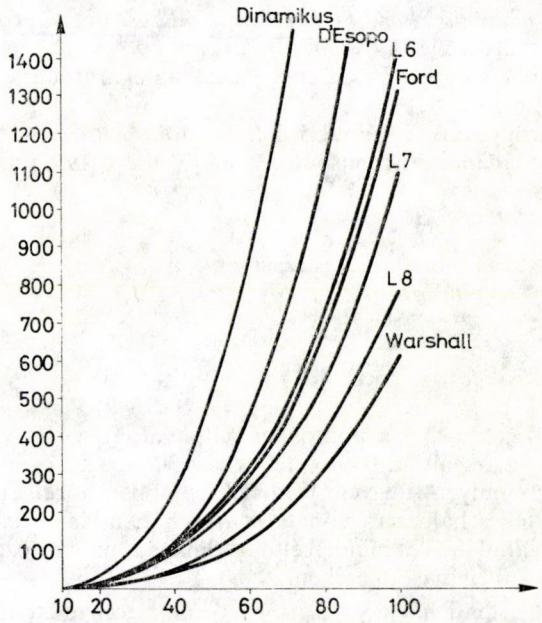
2. TÁBLÁZAT

Futási idők tömör hálózaton

Módszer \ Pontok	10	20	40	60	80	100
Warshall	0,7	8,4	41	137	321	624
Dinamikus	3,1	28	260	865	2375	4630
Ford	1,4	10,9	85	386	672	1317
D'Esopo	1,6	14,3	128	467	1148	2281
L6	1,4	11,5	93	322	743	1452
L7	1,5	11,0	81	263	542	1082
L8	1,8	8,3	54,8	174	403	773

A második legjobb eredményt az L8 módszer szolgáltatta. Ehhez közelálló eredményt adott a módszerünk L7 változata. Ezután sorrendben *Ford* L6, és *D'Esopo módszer* következik. Legrosszabb eredményt — mint az előre sejthető volt — a dinamikus programozási módszer szolgáltatta. A futási idők pontok számától való függését módszerenként felrajzoltuk tömör hálózatok esetén (5. ábra).

A függvények képéből és az 1. táblázatból sejthető, hogy a pontok száma és futási idők közötti összefüggés köbös függvénnyel közelíthető a legjobban. Legkisebb



5. ábra
Futási idők tömör hálózatoknál

négyzetek módszerével a_1x^2, a_2x^3 és a_3x^4 polinómokkal közelítettük az 5. ábrán bemutatott függvényeket. A kapott eredmények igazolták a fenti elképzelést.

A 3. táblázatba foglaltuk össze a megfelelő a_1, a_2 és a_3 együtthatókat és az abszolútérték eltérést. Mint az elméleti becslések is mutatták a köbös alakhoz legközelebb a *Warshall* és a *Ford eljárás* van. Az együtthatók a futási időknek megfelelően a fentiekben elmondott sorrendet követik: legkisebb a *Warshall módszernél*, ezt követ az L8 módszer, majd sorrendben az L7, *Ford*, L6, *D'Esopo* és *Dinamikus módszer*. A tiszta négyzetes és negyedfokú polinóm közelítés mindegyik módszernél nagyságrendekkel nagyobb eltérést adott a köbös alaknál.

3. TÁBLÁZAT

Pontszám — futási idő összefüggés függvény közelítése

Módszer \ Függvény	a_1	Eltérés	a_2	Eltérés	a_3	Eltérés
Warshall	0,5653	237,6	0,00062	5,4	0,000007	164,3
Dinamikus	0,4152	1870,9	0,00461	214,0	0,000048	993,6
Ford	0,1199	511,6	0,00132	5,5	0,000014	335,0
D'Esopo	0,2045	951,8	0,00227	70,9	0,000024	567,4
L6	0,1314	557,1	0,00145	10,4	0,000015	377,2
L7	0,0981	389,6	0,00108	56,6	0,000011	306,6
L8	0,0707	278,9	0,00078	23,5	0,000008	220,7

Mind a futási időket, mind a függvényközelítést elvégeztük különböző sűrűségű hálózatokra is. Kihagytuk az analízisből a mátrix módszereket, mivel ezeknél a hálózat sűrűsége a futási időt nem befolyásolja lényegesen. A futási idők és sűrűség és pontszám együttes értékelésére 31 különböző hálózatot generáltunk.

A futási idők viszonya 50%-os sűrűségű hálózatok esetén lényegében megegyezik a tömör hálózatokéval (lásd I. Melléklet). A különbség csupán annyi, hogy a *Ford módszer* néhány hálózat esetén (20, 40, 60 pontos) megelőzi az L7 módszert. 30%-os sűrűségű hálózatoknál a helyzet hasonló az 50%-os sűrűségű hálózatokhoz: az L7 módszert a *Ford módszer* itt is megelőzi a legtöbb esetben. Ezt tükrözik az I. Mellékletben megadott táblázatok is. Hasonlóan a tömör hálózatokhoz, itt is köbös alak illeszkedik legjobban a kapott futási időkre.

Az a_2 együttható sorrendje a tömör hálózatokéhoz képest csak abban tér el, hogy a *Ford módszerhez* tartozó együttható a 2. értékes jegyben két tizeddel kisebb értéket adott!

A 10%-os sűrűségű hálózatoknál a helyzet némileg módosul. Az L8 módszer mindegyiknél hatékonyabb; a *D'Esopo* a második a 40, 60, 80 csomópontos hálózatok esetén; lényegében azonos időket adott a *Ford* és az L7, és a leghosszabbnak az L6 bizonyult. A 100 pontos 10%-os hálózat esetén a futási idők szempontjából növekvő sorrendben L8, L7, *Ford, D'Esopo* és L6.

Úthossz és útvonal összefüggés

Közlekedési alkalmazások esetén sokszor csak az útvonal hosszára vagyunk kíváncsiak és magára az útvonalra nem (forgalomelőrebecslés gravitációs vagy versengő lehetőségek módszerénél stb.). Ezért meg kell vizsgálnunk, hogy módszerenként mennyivel nő a szükséges gépidő, ha az útvonalat is meg akarjuk határozni. Azaz mennyivel csökken a futási idő, ha csak a leggazdaságosabb útvonal utazási idejének meghatározása szükséges. E célból egy sor azonos hálózaton módszerenként meghatároztuk a két feladathoz tartozó futási időket. A részletes futási eredményeket a II. Melléklet tartalmazza a *Ford* és *D'Esopo módszer* esetén. Teljes hálózatra mátrix módszerekkel is kiszámoltuk az időket a *Warshall* és *dinamikus módszert* egy táblázatba foglaltuk (l. II. Melléklet). Az elemzéshez jobban felhasználható az útvonal meghatározásához szükséges %-os időnövekedés az úthosszhoz viszonyítva (ez utóbbit tekintve 100%-nak). Ezt foglaltuk össze a 4. táblázatban különböző pontszámú hálózatok esetén.

4. TÁBLÁZAT

Út — útvonal futási idejének növekedése %-ban

Módszer \ n	10	20	40	60	80	100
Warshall	40	54,2	51,8	45,0	50,0	50,0
Dinamikus	25,9	47,4	49,4	48,3	48,7	48,8
Ford	7,7	5,8	4,7	5,5	4,8	5,4
D'Esopo	6,7	5,9	4,9	5,9	5,0	5,3

A mátrix módszereknél az időnövekedés mintegy 50%, míg a fa építő algoritmusoknál ez az úthossz meghatározási idejéhez képest elenyésző (átlag 5%). Így csak útvonal meghatározásához, ha memória nem korlátoz, a *Warshall eljárást* célszerű használni. Ez abszolút értelemben is sokkal jobb, mint a legjobb faépítő módszer sűrű hálózatokra. Az 50%-nál ritkább hálózatok esetén azonban az L8 módszer jobb eredményt ad. Így 30%-os hálózatnál az L8 ideje 100 pontos hálózatra 278 mp, míg a *Warshall módszeré* 416 mp. Megvizsgáltuk, hogy a hálózat sűrűsége hogyan befolyásolja az útvonal meghatározás idejét. A *Ford módszere* a hálózat sűrűségétől függetlenül 3,64—6,06% közötti időnövekedést adott. Hasonló eredmény adódott a *D'Esopo módszer* esetén is (4,91—6,22%). Így a hálózat sűrűsége nincs jelentős befolyással az útvonal meghatározásához szükséges időnövekedésre (1. II. Melléklet).

Élek számának hatása a futási időre

A módszer egyik jelentős sajátossága a futási idő növekedése a hálózat sűrűségének (élek számának) növekedésével. E célból különböző pontszámú és sűrűségű hálózatokon futtattuk le a módszereket. Ezt mutatjuk be a III. Mellékletben. Száz csomópontból álló hálózat futási idő növekedését különböző sűrűségű hálózatokban az 5. táblázat tartalmazza. Itt a kisebb sűrűségű hálózat futási idejét 100%-nak véve kiszámoltuk a nagyobb sűrűségű hálózat idejének %-os növekedését.

5. TÁBLÁZAT

Futási idő %-os növekedése a hálózat sűrűség növekedésével

Módszer / Sűrűség v	*Ford*	*D'Esopos*	L6	L7	L8
5— 10%	9,2	95,7	19,7	68,0	39,8
10— 30%	32,3	86,1	24,4	34,3	126,0
30— 50%	24,1	31,7	14,5	6,4	51,8
50— 80%	30,2	30,9	11,7	18,0	49,5
80—100%	14,9	20,2	10,7	19,7	22,5
10—100%	145,7	285,9	85,7	92,2	582,4

A táblázatból látszik, hogy a ritka hálózat esetén csekély sűrűségnövekedés relatíve sokkal nagyobb futási időnövekedést okoz, mint sűrű hálózat esetén. Így például 5%-ról 10%-ra való sűrűség növekedés a *Ford módszernél* 9,2%-os futási időnöveke dést okoz, míg 80%-os sűrűségről 100%-osra való sűrűség növekedés csak 14,9%-os időnövekedéssel jár. A sűrűségre legérzékenyebb az L8 módszer, ezután következik a *D'Esopo módszer*, majd sorrendben a *Ford*, L7 és L6 módszerek. Ezt összevetve az abszolút futási időkkel azt kapjuk, hogy az L8 és a *D'Esopo módszert* a legcélszerűbb használni ritka hálózatok esetén.

Egészen ritka négyszög hálózaton azonban jobb eredményt kaptunk módszerünkkel (L7), mint a *D'Esopo módszerrel*. Ezt bizonyítják a következőkben bemutatandó négyszög hálózat kapott futási idők is.

A III. Mellékletből az az érdekes összefüggés adódik, hogy

— az L6, L7 és a *D'Esopo módszer* esetén minél nagyobb pontszámú hálózatnál növeljük a hálózat sűrűségét, annál kisebb időnövekedést kapunk;
— az L8 módszernél a helyzet éppen fordított — azonos sűrűség növekedés nagyobb pontszámú hálózatok esetén relatíve nagyobb időnövekedést okoz, mint kisebb pontszámú hálózatoknál;
— a *Ford módszer* esetén a relatív időnövekedés a pontszámtól független.

Legkisebb négyzetek módszerével 100, 80 és 60 csomópontos hálózatok esetén, több függvénytípussal közelítettük a sűrűség-futási idő függvényt. A következő alakú függvényeket próbáltuk ki: $ax+b$, $\ln x$, $x^{2/3}$, x^2, a $\ln x+bx$. Némely módszernél megfelelő eredményt kaptunk lineáris és az a $\ln x+bx$ alakú függvényekkel. A legkisebb négyzetek módszerével kapott paramétereket, és a számított és tényleges függvényértékek közötti eltérések abszolút értékeinek összegét a IV. Mellékletben foglaltuk össze. A legjobb abszolútérték eltéréseket adó függvények paramétereit 100 pontos hálózatra a 6. táblázat tartalmazza. A 80 és 60 pontos hálózatra kapott értékeket a fenti mellékletben közöljük.

A függvényközelítés eredménye, hogy a *Ford* és az L8 *módszerek* esetén igen jól közelít a lineáris függvény. Az L6 módszer esetén lineáris, az L7 és *D'Esopo módszerek* esetén pedig az $a \ln x+bx$ alak adott a legnagyobb eredményt, de viszonylag nagy eltérések adódtak.

6. TÁBLÁZAT

Sűrűség növekedési függvény közelítés 100 pontos hálózatnál

Módszer \ fv.	$ax+b$		eltérés	$a \ln x+bx$		
	a	b		a	b	eltérés
Ford	8,7	447,8	7,4	201,2	3,4	400,0
D'Esopo	19,6	317,7	521,8	189,3	13,9	204,6
L6	7,9	684,5	205,5	316,5	−0,7	461,5
L7	6,7	414,0	553,5	199,2	1,0	263,1
L8	7,2	55,4	24,9	25,8	6,5	30,2

7. TÁBLÁZAT

Futási idők négyszög hálózaton

Módszer \ Pontok	10×10	10×20	10×30	20×30	20×40
Ford	496	3569	11 836	92 986	219 281
D'Esopo	143	1021	3 688	29 735	84 978
L6	548	4022	131 138	103 380	241 744
L7	218	960	2 265	13 790	25 754
L8	105	395	874	2 843	4 982
Élek száma	180	370	560	1 150	1 540
Sűrűség (%)	1,82	0,93	0,62	0,32	0,24

Négyszög hálózatok

A fa építő módszereket négyszög hálózatokon is kipróbáltuk. A futási időket különböző hálózatok esetén a 7. táblázatban adjuk meg. A fenti megállapításokkal egyezően azt az eredményt kaptuk, hogy a pontszám növekedéssel és az egyidejű sűrűség csökkenéssel a leghatékonyabb eljárás az L8 módszer. Jó eredményt szolgáltat még az L7 módszerünk. A *D'Esopo*, a *Ford* és az L6 *módszer* az előző módszereknél egy nagyságrenddel hosszabb futási időt ad.

I. MELLÉKLET

Leggazdaságosabb útvonalak futási ideje a sűrűség függvényében

L6 módszer

Sűrűség % \ n	10	20	40	60	80	100
5	—	—	—	110	316	653
10	—	—	43	157	389	782
30	0,8	7,4	61	207	496	973
50	1,0	8,8	72	241	568	1114
80	1,3	10,6	84	295	675	1311
100	1,4	11,5	93	322	743	1452

L7 módszer

Sűrűség % \ n	10	20	40	60	80	100
5	—	—	—	21	128	319
10	—	—	22	106	269	536
30	0,5	7,0	58	174	393	720
50	0,9	8,8	65	218	409	766
80	1,1	10,8	70	263	519	904
100	1,5	11,0	81	269	542	1082

L8 módszer

Sűrűség % \ n	10	20	40	60	80	100
5	—	—	—	89	55	88
10	—	—	16,1	35	70	123
30	1,4	44,9	23,7	68	49	278
50	1,5	5,6	32,6	99	223	422
80	1,6	7,3	46,0	144	332	631
100	1,8	8,3	54,8	174	403	773

Az utazási idők és címkék meghatározása a sűrűség függvényében

Ford módszere

Sűrűség % \ n		10	20	40	60	80	100
5%	U	—	—	—	102	241	468
	C	—	—	—	107	254	491
10%	U	—	—	33	111	262	510
	C	—	—	35	117	277	536
30%	U	0,7	5,7	44	147	346	674
	C	0,8	6	46	155	364	709
50%	U	0,9	7	55	182	431	838
	C	1,0	7,4	57	193	452	880
80%	U	1,2	9	70	236	557	1085
	C	1,2	9,5	74	249	585	1146
100%	U	1,3	10,3	81	271	641	1249
	C	1,4	10,9	85	286	672	1317

Táblázatban használt jelölések:
U — csak úthossz meghatározás
C — útvonal és úthossz meghatározás

D'Esopo módszere

Sűrűség % \ n		10	20	40	60	80	100
5	U	—	—	—	13	113	286
	C	—	—	—	13	120	302
10	U	—	—	17	84	251	559
	C	—	—	18	89	265	591
30	U	0,4	4,6	53	201	495	1045
	C	0,4	4,8	56	213	522	1100
50	U	0,7	7,0	72	269	678	1372
	C	0,8	7,5	76	285	714	1449
80	U	1,1	10,8	100	358	916	1786
	C	1,2	11,4	106	379	963	1897
100	U	1,5	13,5	122	441	1093	2164
	C	1,6	14,3	128	467	1148	2281

II. MELLÉKLET

Utazási idők és útvonalkiválasztás idejének összehasonlítása

Teljes hálózaton mért futási idők

Módszer \ n	10	20	40	60	80	100
Warshall U	0,5	3,5	27	91	214	416
Warshall C	0,7	5,4	41	137	321	624
Dinam U	2,7	19	174	580	1597	3111
Dinam C	3,1	28	260	865	2375	4630
Ford U	1,3	10,3	81	271	641	1249
Ford C	1,4	10,9	85	286	672	1317
D'Esopo U	1,5	13,5	122	441	1093	2164
D'Esopo C	1,6	14,3	128	467	1148	2281

A táblázatban az U jelöli az utazási idők, C az útvonalak meghatározásához szükséges időket.

Az útvonal meghatározásához szükséges időnövekedés %-ban az utazási időkhöz képest — különböző sűrűségű és pontszámú hálózatok esetén.

Ford módszer

% \ n	40	60	80	100
5	—	4,90	5,39	4,91
10	6,06	5,41	5,72	5,10
30	4,54	5,44	5,20	5,19
50	3,64	6,04	4,87	5,01
80	5,71	5,51	5,03	5,44
100	4,93	6,54	4,84	5,44

D'Esopo módszer

% \ n	40	60	80	100
5	—	0	6,19	5,6
10	5,88	5,95	5,58	5,72
30	5,67	5,97	5,45	6,22
50	5,56	5,98	5,31	5,61
80	6,00	5,87	5,13	6,21
100	4,91	5,90	5,03	5,31

III. MELLÉKLET

Futási idők függése a sűrűségtől

Futási idők százalékos növekedése különböző sűrűségváltozások és hálózatok esetén

Sűrűség változás	*n*	*Ford*	*D'Esopo*	L6	L7	L8
5— 10%	100	9,2	95,7	19,7	68,0	39,8
	80	9,0	120,8	23,1	110,1	27,3
	60	9,3	584,6	42,7	404,8	293,8
10— 30%	100	32,3	86,1	24,4	34,3	126,0
	80	31,4	97,0	27,5	46,1	112,8
	60	32,5	139,3	31,8	64,1	94,3
	40	31,4	211,1	41,8	163,6	47,2
30— 50%	100	24,1	31,7	14,5	6,4	51,8
	80	24,2	36,8	14,5	4,1	49,7
	60	24,5	33,8	16,4	25,3	45,6
	40	23,9	35,7	14,7	12,1	33,5
	20	23,3	56,2	18,9	25,7	14,3
50— 80%	100	30,2	30,9	17,7	18,0	49,5
	80	29,4	34,9	18,8	26,9	48,9
	60	29,0	33,0	22,4	20,6	45,4
	40	29,8	39,5	16,7	7,7	41,1
	20	28,4	52,0	20,4	22,7	30,3
80—100%	100	14,9	20,2	10,7	19,7	22,5
	80	14,9	19,2	10,0	4,4	21,4
	60	14,8	23,2	9,1	1,1	20,8
	40	14,9	20,7	10,7	15,7	19,1
	20	14,7	25,4	8,5	1,9	13,7
10—100%	100	145,7	285,9	85,7	92,2	582,4
	80	142,6	333,2	91,0	101,5	475,7
	60	144,4	424,7	105,0	153,8	397,1
	40	142,8	611,1	116,2	268,1	240,4

IV. MELLÉKLET

Függvényközelítés futási idők — sűrűségnövekedés összefüggésre

100 csomópontos hálózat esetén

Függvény / Módszer	$ax+b$			$a \ln x$		$a x^3$		ax^2		$ax+b \ln x$		
	a	*b*	eltérés	*a*	eltérés	*a*	eltérés	*a*	eltérés	*a*	*b*	eltérés
Ford	8,7	447,8	7,4	252,3	581,5	1,6	2015,8	0,25	2456,8	201,2	3,4	400,0
D'Esopo	19,6	317,7	521,8	400,0	1631,3	2,6	2335,8	0,36	3078,0	189,3	13,9	204,6
L6	7,9	684,5	205,5	305,3	459,2	1,8	2887,3	1,98	3394,5	316,3	−0,7	461,5
L7	6,7	414,1	553,5	214,7	282,2	1,3	1838,2	0,12	2172,7	199,2	1,0	263,1
L8	7,2	55,3	24,9	124,3	771,2	0,9	532,7	0,09	775,6	25,8	6,5	30,1

80 csomópontos hálózat esetén

Függvény / Módszer	$ax+b$			$a \ln x$		ax^3		ax^2		$ax+b \ln x$		
	a	*b*	eltérés	*a*	eltérés	*a*	eltérés	*a*	eltérés	*a*	*b*	eltérés
Ford	4,4	232,2	2,7	129,2	291,39	0,8	1040,7	0,080	1266,0	104,4	1,6	207,3
D'Esopo	10,3	149,9	248,4	197,9	932,95	1,3	1073,5	0,13	1446,1	78,3	7,9	96,4
L6	4,2	377,2	120,9	155,5	2203,31	0,9	1448,1	0,92	1707,9	156,9	−0,1	203,6
L7	3,9	203,2	267,6	113,5	145,35	0,7	988,1	0,68	1181,1	102,5	0,7	131,6
L8	3,7	36,4	10,5	65,7	385,05	0,4	302,1	0,05	429,5	16,6	3,2	24,1

60 csomópontos hálózat esetén

Függvény / Módszer	$ax+b$			$a \ln x$		ax^3		ax^2		$ax+b \ln x$		
	a	*b*	eltérés	*a*	eltérés	*a*	eltérés	*a*	eltérés	*a*	*b*	eltérés
Ford	1,9	98,2	2,3	55,0	123,1	0,3	441,6	0,03	537,6	44,2	0,7	85,4
D'Esopo	4,4	38,5	147,6	78,1	420,9	0,5	366,5	0,05	516,0	23,6	3,6	95,7
L6	2,1	127,3	79,1	66,0	67,4	0,4	578,7	0,04	691,0	60,5	0,4	56,3
L7	2,3	68,4	182,8	54,7	143,3	0,3	419,0	0,34	515,6	38,4	1,1	102,3
L8	1,6	12,6	31,6	128,5	171,0	0,2	127,1	0,02	182,4	6,9	1,4	17,4

IRODALOM

[1] "ADECODE, Scientific Models of Traffic", METRA International, 1972.
[2] ASSAD, A. A., "Multicommodity network flows — a survey", *Networks* **8** (1978) 37—91.
[3] BAKÓ, A., „Minimális és multiterminális minimális út feladat megoldása veszteséges hálózatban". Doktori disszertáció, ELTE, 1971.
[4] BAKÓ, A., „Multiterminális minimális út feladat megoldási módszerei és alkalmazásai", *MTA SZTAKI Közlemények* **6** (1971) 49—54.
[5] BAKÓ, A., „Nagyméretű közúti forgalomtervezési feladat megoldása számítógéppel", *BME—KAMM* **21** (1976) 47—65.
[6] BAKÓ, A., "Some problems and algorithms for traffic planning and forecasting", *Mathematische Gesellschaft der DDR* **1** (1976) 11—15.
[7] BAKÓ, A., „Ritka mátrixok számítógépes tárolási módszerei", *KTMF Tudományos Közlemények* **2** (1978) 351—357.
[8] BAKÓ, A., „Forgalomelosztási módszerek fejlesztése és alkalmazása nagyvárosok közlekedési hálózatának tervezéséhez", *Városi Közlekedés* **19** (1979) 281—287.
[9] BELLMAN, R., "On a routing problem", *Q. of Applied Mathematics* **16** (1958) 87—90.
[10] BÉNYEI, A. és PARLAGI, E., „A városi úthálózat várható forgalmának meghatározása a kapacitás-viszonyokkal számoló program alapján", *Közlekedéstudományi Szemle* **22** (1972) 503—509.
[11] BURELL, J. E., "Multiple route assignment and its application to capacity restraint", *4th International Symp. Theory of Traffic Flow*, Karlsruhe, 1968.
[12] BURELL, J. E., "Multiple route assignmenet: A comparision of two methods", *Traffic Equilibrium Methods* (1976) 229—239.
[13] CHARNES, A. and COOPER, W. W., "Multicopy traffic network models", *Proceedings of the Symposium on the Theory of Traffic Flow*, R. Herman P. C., 1961.
[14] DAFERMOS, S. C. and SPARROW, F. T., "The traffic assignment problem for a general network", *Journal of Research of NBS—B.* **73** (1969) 91—118.
[15] DIAL, R., "Algorithm 360, Shortest path forest with topological ordering", *Communication of the ACM* **19** (1965) 632—633.
[16] DIAL, R. B., "A probabilistic multipath traffic assignment model wich obviates path enumeration", *Transportation Research* **5** (1971) 83—111.
[17] DIJKSTRA, E. W., "A note on two problems in connection with graphs", *Numerische Mathematik* **1** (1959) 269—271.
[18] DREYFUS, S. E., "An appraisal of some shortest path algorithms", *Operations Research* **17** (1969) 395—412.
[19] EDMONDS, J. and KARP, R. M., "Theoretical improvements in algorithmic efficiency for network flow problems", *Journal of Association for Computing Machinery* **19** (1972) 248—264.
[20] ERLANDER, S., "Accessibility, entropy and the distribution and assignment of traffic", *Transportation Research* **11** (1977) 149—153,
[21] EVANS, S. P., "Some models for combining the trip distribution and traffic assignment stages in the transport planning process", *Traffic Equilibrium Methods* (1976) 201—228.
[22] FHWA Computer Programs for Urban Transportation Planning, U.S. Dep. of Transp., Fed. Higlw. Adm. 1974.
[23] FLORIAN, M. and NGUYEN, S., "Combined trip distribution model split and trip assignment model", Universite De Montreal, 1977.
[24] FLOYD, R. W., "Algorithm 97, Shortest path", *Communication of ACM* **5** (1962) 345.
[25] FORD, J. R., "Network flow theory", The RAND Corporation, Paper 923, 1956.
[26] FRANK, M. and WOLFE, P., "An algorithm of quadratic programming", *Naval Research Logistics Quarterly* **3** (1956) 95—110.
[27] FULKERSON, D. R., "An out-of-kilter method for minimal cost flow problems", *SIAM Journal* **9** (1961) 18—27.
[28] GOLDEN, B., "Shortest-path algorithms: A comparision", *Operations Research* **24** (1976) 1164—1168.
[29] HALL, M. A. and PETERSON, E. L., "Traffic equilibria analysed via geometric programming", International Symposium on Traffic Equilibrium Methods, 1974, Montreal, Discussion Paper 130.
[30] HOLM, J., JENSEN, T. and NIELSEN, S. K., "Calibrating traffic models on traffic census results only", *Ttaffic Engineering and Control* (1976) 137—140.
[31] HOROWITZ, A. J., "The subjective value of the time spent in travel", *Transportation Research* **12** (1978) 85—93.

[32] HU, T. C., "Revised matrix algorithms for shortest paths", *SIAM J. Appl. Math.* **15** (1967) 207—218.
[33] IRI, M., "Principle of incremental assignment — largescale problem on small computer", IIASA Simposium on Solving Larga-scale Math. Prog. Problem, Luxemburg, Nov. 4—6, 1974.
[34] KALABA, R., "On some communication network problems", Combinatorial Analysis, Proc. Symp. Appl. Math., (1960) 261—280.
[35] KAS, P. and MAYER, J., "A reduce gradient approach to the nonlinear network flow problem", Working Paper of CAI, 1979.
[36] KENNINGTON, J. L., "A survey of linear cost multicommodity network flows", *Operations Research* **26** (1978) 209—236.
[37] KIRÁLY, L. és BAKÓ, A., "DECOMP program és használata", MTA SZTAKI Kutatási Jelentés, 1975.
[38] KLAFSZKY, E., *Hálózati folyamok* (Bolyai János Matematikai Társulat Budapest, 1969).
[39] KLEIN, M., "A primal method for minimal cost flow with application to the assignment and transportation problem", *Management Science* **24** (1967) 1—34.
[40] KOLLER, S., *Forgalomtechnika,* (Tankönyvkiadó, Budapest 1976).
[41] LEBLANC, L. J., "An analysis and comparision of behavioral assumption in traffic assignment", *Traffic Equilibrium Methods* (1976) 413—425.
[42] LUENBERG. D. G., *Introduction to Linear and Nonlinear Programming,* (Addision-Wisley, P. C., 1973).
[43] MANDL, C., *Applied Network Optimization,* (Academic Press, London, 1979).
[44] MICHAELS, R. M., "Attitudes of drivers determine choice between alternate highways", *Public Roads* **33** (1965) 225—236.
[45] MINTY, G. J., "A comment on the shortest route problem", *Operations Research* **5** (1957) 724—728.
[46] MONIGL, J., „Forgalomráterhelés számítógépes módszerének továbbfejlesztése", 221—72/02—02 KKTKI Tanulmány.
[47] MONIGL, J. és VÁSÁRHELYI, B., „Analitikus forgalomelőrebecslési módszerek vizsgálata", *KÖTUKI Kiadványok* **20** (1975).
[48] MORRIS, R. J., "A comparative analysis of trip distribution and traffic assignment models for transportation plann.ng in developing regions", Stanford University, 1973.
[49] MURCHLAND, J. D., "Method of diverted assignment over adjacent routes used in London underground programs", Report LBS-TNT-102. l., London School of Business Studies.
[50] NAGY, K. és MARTON, M., „Kapacitáskorlátos forgalomráterhelés az úthálózatra", *Közlekedés Tervezése Elektronikus Számítógéppel* **4** (1975) 1—67.
[51] NGUYEN, S., "Mathematical programming approach to equilibrium methods of traffic assignment with fixed demands", Universite de Montreal 1973.
[52] NGUYEN, S., "An algorithm for the traffic assignment problem", *Transportation Science* **8** (1974) 203—213.
[53] PIGOU, A. C., *The Economics of Welfare,* (Macmillian, London, 1920).
[54] PITFIELD, D. E., "Sub-optimality in freight distribution", *Transportation Research* **12** (1978) 403—409.
[55] Planpac/Backpac, "Computer Programs for Urban Transportation Planning", U. S. Dep. of Trans. Federal Highway Administration, 1976.
[56] POLLACK, M. and WIEBENSON, W., "Solution of the shortest route problem — A review", *Operations Research* **8** (1960) 224—230.
[57] POTTS, R. B. and OLIVER, R. M., *Flow in Transportation Network,* (Academic Press, 1972).
[58] RUITER, E. R., "Network equilibrium capabilities for the UMTA transportation planning system", Traffic Equilibrium Methods, 1976, Springer V., 183—199.
[59] SHIMBEL, A., "Structure in communication nets", Proc. of the Symposium on Information Network, Booklin Politechn. Inst. New York, 1954.
[60] TOMLIN, J. A., "Minimum-cost multicommodity network flows", *Operations Research* **14** (1966) 45—51.
[61] TRANPLAN Transportation Planning System, CDC 1971.
[62] Urban Transportation Planning General Information, U.S. Department of Transportation, Federal Highway Administration, 1972.
[63] VENUS-Programmsystem zur Berechnung der Verkehrsverteilung in Strassennetzen, SIMENS, 1970.
[64] VLIET, D. V., "Road assignment II: The GLTS model", *Transportation Research* **10** (1976) 145—149.

[65] VLIET, D. V., "Improved shortest path algorithms for transport networks", *Transportation Research* **12** (1978) 7—20.
[66] VLIET, D. V., "Road assignment III: Comparative test of stochastic methods", *Transportation Research* **10** (1976) 151—157.
[67] WACHS, M., "Relationship between driver's attitudes toward alternate routes and driver and route characteristics", *Hyghway Research Bulletin Record* **197** (1967) 70—87.
[68] WARDROP, J. G., "Some theoretical aspects of road traffic research", *Proceedings Institute of Civil Engeneering* **36** (1952) 325—378.
[69] WARSHALL, S., "A theorem on boolean matrices", *Journal of ACM* **9** (1962) 11—12.
[70] WHITMAN, D., "Computerized models for Brazilian highway master planning", SYSTAN Inc. 1971, SPR 71—11.
[71] ZOUTENDIJK, G., *Methods of Feasible Directions,* (Elsevier Publishing Company, Amsterdam and New York, 1960).

(Beérkezett: 1980. december 15.)

BAKÓ ANDRÁS
KÖZLEKEDÉSI ÉS TÁVKÖZLÉSI MŰSZAKI FŐISKOLA
9026 GYŐR, SÁGVÁRI E.U. 3.

TRAFFIC ASSIGNMENT BY COMPUTER

A. BAKÓ

The most important step of traffic planning and forecasting is the network distribution of the traffic.

Assignment models can be defined two ways. Both of them result an optimization problem. These problems usually can be solved by heuristic methods because of the large size of the traffic network. Both the exact and the heuristic procedures apply the shortest path algorithms. In the paper we give a summary of the important traffic assignment and shortest path procedures. We give some computational experiences, too.

Alkalmazott Matematikai Lapok 6 (1980) 393—406

EGY ÚJ MODELL RÚDSZERKEZETEK OPTIMÁLIS MÉRETEZÉSÉRE

BERNAU HEINZ
Budapest

HALMOS EMIL
Győr

SOÓS ZSOLT
Budapest

Rúdszerkezetek méretezési feladatainál a feltételi függvények általában csak implicit módon állnak elő. A megoldást valamilyen matematikai programozási algoritmussal szeretnénk megkapni. Ennek érdekében a feltételi függvényeket a tervezési változók egy explicit függvényével közelítjük. E célból a szerkezet egy olyan felbontását adjuk meg, amely segítségével a teljes szerkezet méretezését több, kisméretű, egyszerűbben kezelhető részszerkezet méretezésével helyettesíthetjük. Ezen felbontás által mód nyílik nagyméretű szerkezetek részekre bontására, és ezek egymástól független méretezésére. A felállított modell egy nemlineáris programozási feladat.

1. Bevezetés

Rúdszerkezetek méretezésekor adott külső terhelés esetén több, a funkcionális követelményeknek megfelelő szerkezet közül választhatunk. Így érdemes valamilyen szempontot figyelembe venni és eszerint legjobb szerkezetet kiválasztani. Ilyen szempontok lehetnek a rudak száma, a szerkezet súlya stb.

Az ilyen feladatok matematikai megfogalmazásakor az a probléma merül fel, hogy a szerkezet viselkedését leíró összefüggéseket nem tudjuk explicit módon megadni, mint azt a következő fejezetben meg fogjuk mutatni. E nehézség megoldására két fő irány alakult ki. Az első típusba tartozók nemlineáris módszereket alkalmaznak és minden iterációs lépésben a szükséges értékeket az implicit relációk megoldásával határozzák meg. Ez igen munkaigényes, mert így minden lépésben egyenlőségrendszereket kell megoldani vagy integrálokat kell kiszámítani. A másik típusú módszerek az implicit relációkat explicit relációkkal approximálják és a keletkező problémákra optimalitási tulajdonságokat és kritériumokat állítanak fel, amelyeket iteratív módon oldanak meg. Ezekről a típusokról VENHAYYA [8] adott részletes áttekintést.

A két típus közötti összefüggéseket FLEURY és GERADIN [2], valamint SAUNDER és FLEURY [6] vizsgálták, és egy olyan módszert javasoltak, amelyik a két típus lépéseit kombinálja. A fenti vizsgálatok azt mutatták, hogy az implicit összefüggések kezelése a feladatok effektív megoldására nézve nagyon lényeges.

A dolgozatban a rúderők keresztmetszettől való függésére explicit közelítést adunk. Ez a modell a *Halmos—Rapcsák féle modell* [4], [5] egy módosítása, és a rúdszerkezet speciális felbontásán alapszik.

A következő fejezetben a megoldandó feladatot ismertetjük és néhány általános megjegyzést teszünk, majd pedig röviden ismertetjük a *Halmos—Rapcsák modellt.* A 4. fejezetben az új felbontási elv algebrai levezetését írjuk le, majd pedig számítástechnikai összehasonlító eredményeket közlünk.

2. A feladat megfogalmazása

Rúdszerkezeteket akarunk tervezni, amelyek adott anyagból elkészülve, a rudak elhelyezkedését és egymáshoz kapcsolódási módját rögzítve, az előre megadott külső statikus terhelést elviselik. A rúdszerkezetet úgy határozzuk meg, hogy megadjuk a rudak keresztmetszeteit oly módon, hogy meghatározott mechanikai korlátok figyelembevételével egy adott célfüggvény optimális legyen. A rúdszerkezetre a következő megkötéseket tesszük:

a) a szerkezet rúdjai állandó keresztmetszetűek;
b) a külső terhelések kizárólag csak a csomópontokban hatnak, és kinematikai terhelés nem hat;
c) a csomópontok elmozdulásai differenciálisan kicsik és a hatásuk az erőegyensúlyra elhanyagolható; (elsőrendű elmélet)
d) a rudak ideálisan elasztikusak;
e) a rúdszerkezet statikailag határozatlan.

A fenti feltételezések mellett a szerkezet viselkedését a következő egyenlőségrendszer írja le [7] [3]:

$$\begin{aligned} \mathbf{R}(\mathbf{t})\mathbf{y} - \mathbf{A}^T\mathbf{x} &= \mathbf{0} \\ \mathbf{A}\mathbf{y} &= \mathbf{q}, \end{aligned} \tag{2.1}$$

ahol

$\mathbf{R}(\mathbf{t})$ — a szerkezet rugalmassági mátrixa, amely a rudak keresztmetszeti jellemzőinek (továbbiakban keresztmetszetek) $\mathbf{t}=(t_1, t_2, \ldots, t_N)$ függvénye.
$\mathbf{A}$ — a szerkezet geometriai mátrixa, amely a szerkezetre jellemző konstansokat tartalmazza, mint pl. rúdhosszak, iránykoszinuszok stb.
$\mathbf{A}^T$ — az $\mathbf{A}$ mátrix transzponáltja
$\mathbf{y}$ — a rúderők vektora
$\mathbf{x}$ — a csomóponti elmozdulások vektora
$\mathbf{q}$ — a külső terhelés vektora

Abban az esetben, ha súlyminimális szerkezetet akarunk meghatározni a rudakban ébredő feszültségek korlátozása mellett, a következő alakú probléma áll elő:

$$\min_{(t_1, t_2, \ldots, t_N)} \varrho \sum_{i=1}^{N} l_i t_i \tag{2.2}$$

$$\sigma_i(\mathbf{y}, \mathbf{t}) \leqq \bar{\sigma}_i, \quad i = 1, 2, \ldots, N \tag{2.3}$$

$$t_i \geqq \bar{t}_i, \quad i = 1, 2, \ldots, N, \tag{2.4}$$

ahol l_i az i-edik rúd hossza, ϱ a szerkezet anyagának fajsúlya és $\sigma_i(\mathbf{y}, \mathbf{t})$ az i-edik rúdban levő feszültség, ami az $\mathbf{y}$ rúderők és a $\mathbf{t}$ keresztmetszetek függvénye. A $\bar{\sigma}_i$ és $\bar{t}_i$ konstansok az i-edik rúdban ébredő feszültség felső korlátja, valamint az i-edik rúd keresztmetszetének megengedett minimális értéke.

Az ilyen problémák nehézsége az, hogy előre megadott $\mathbf{t}$ keresztmetszetvektor esetén az $\mathbf{y}$ rúderővektort csak a (2.1), az ún. alapegyenletrendszer megoldásával tudjuk megkapni.

Mivel az $\mathbf{R}(\mathbf{t})$ rugalmassági mátrix $\mathbf{t}>\mathbf{0}$ vektorok esetén reguláris, a (2.1) rendszerből kapjuk:

$$\mathbf{y}(\mathbf{t}) = \mathbf{R}^{-1}(\mathbf{t})\,\mathbf{A}^T\mathbf{x}, \tag{2.5}$$

amelyből a (2.1) alaprendszer második egyenletének felhasználásával a következő áll elő:

$$\mathbf{A}\mathbf{y} = \mathbf{A}\mathbf{R}^{-1}(\mathbf{t})\,\mathbf{A}^T\mathbf{x} = \mathbf{q}.$$

Mivel a szerkezet statikailag határozatlan, így az $\mathbf{A}$ mátrix teljes sorrangú, így a

$$\mathbf{C}(\mathbf{t}) = \mathbf{A}\mathbf{R}^{-1}(\mathbf{t})\,\mathbf{A}^T$$

mátrix invertálható. Így az $\mathbf{x}$ csomóponti elmozdulásvektor a külső terhelés segítségével kifejezhető. Azaz:

$$\mathbf{x} = [\mathbf{A}\mathbf{R}^{-1}(\mathbf{t})\,\mathbf{A}^T]^{-1}\mathbf{q} = \mathbf{C}^{-1}(\mathbf{t})\mathbf{q}.$$

Ezt a kifejezést (2.5)-be behelyettesítve az $\mathbf{y}$ erővektor következő előállítását kapjuk:

$$\mathbf{y}(\mathbf{t}) = \mathbf{R}^{-1}(\mathbf{t})\,\mathbf{A}^T[\mathbf{A}\mathbf{R}^{-1}(\mathbf{t})\,\mathbf{A}^T]^{-1}\mathbf{q} = \mathbf{R}^{-1}(\mathbf{t})\,\mathbf{A}^T\mathbf{C}^{-1}(\mathbf{t})\mathbf{q}. \tag{2.6}$$

Így a (2.2), (2.3), (2.4) optimalizálási probléma végül a következő alakot ölti:

$$\begin{gathered}\min_{(t_1,\ldots,t_N)} \varrho \sum_{i=1}^{N} l_i t_i \\ \sigma_i(\mathbf{y}(\mathbf{t}), \mathbf{t}) \leqq \bar{\sigma}_i, \quad i = 1, 2, \ldots, N \\ t_i \geqq \bar{t}_i, \quad i = 1, 2, \ldots, N,\end{gathered} \tag{2.7}$$

ahol az $\mathbf{y}(\mathbf{t})$ a (2.6) alapján nyerjük. Ebből a relációból látható, hogy az $\mathbf{y}(\mathbf{t})$ explicit módon nem meghatározható, mivel a $\mathbf{t}$ vektor a $\mathbf{C}(\mathbf{t})$ mátrix inverzében is fellép. Megjegyezzük, hogy az $\mathbf{R}^{-1}(\mathbf{t})$ mátrix az $\mathbf{R}(\mathbf{t})$ mátrix blokkstruktúrája miatt a keresztmetszetek explicit függvényeként megadható.

Az előbb felírt modellben a $\mathbf{t}$ rúdkeresztmetszetvektor és az $\mathbf{y}$ rúderővektor közötti implicit összefüggést egy explicit függvénnyel fogjuk helyettesteni, amely a (2.6)-ban meghatározott $\mathbf{y}(\mathbf{t})$ vektor egy közelítése lesz. Így a módosított feladat megoldása lényegesen egyszerűbbé válik. A megoldás során néhány iterációs lépés után az approximációt lehet, hogy ismételni kell, mivel a tényleges rúderők és a közelítéssel kapott értékek között lényeges eltérés adódik. SAUNDER és FLEURY az $\mathbf{y}(\mathbf{t})$ vektort lineárisan approximálták, míg mi mechanikai meggondolások alapján nemlineáris függvénnyel közelítjük a rúderővektort.

3. A Halmos—Rapcsák modell

Ez a modell rácsos tartók súlyminimális méretezésére készült [4] [5] és később BERNAU és HALMOS [1] ezt olyan szerkezetekre is általánosították, amelyben nem csak normálerőket, hanem nyíróerőket és nyomatékokat is figyelembe vesznek.

Az explicit approximáció érdekében a szerkezetet bontsuk fel részszerkezetekre. A részszerkezeteket azok a rudak alkotják, amelyek egy elmozduló csomópontba futnak össze. A rudak szabad végpontjait rögzítetteknek tételezzük fel. Megjegyezzük

továbbá, hogy bizonyos rudak két részszerkezetben is szerepelnek, ezek azok a rudak, amelyek két elmozduló csomópontot kötnek össze.

A kis szerkezet egyensúlyára is a (2.1)-hez hasonló alaprendszer írható fel:

$$\begin{aligned} \mathbf{R}_\gamma(\mathbf{t})\mathbf{y}_\gamma - \mathbf{A}_\gamma^T \mathbf{x}_\gamma &= \mathbf{0} \\ \mathbf{A}_\gamma \mathbf{y}_\gamma &= \mathbf{q} \end{aligned} \qquad \gamma = 1, 2, \ldots, M. \tag{3.1}$$

A rúderőkre pedig a (2.6)-hoz hasonló alakú kifejezés adódik:

$$\mathbf{y}_\gamma(\mathbf{t}) = \mathbf{R}_\gamma^{-1}(\mathbf{t})\mathbf{A}_\gamma^T[\mathbf{A}_\gamma \mathbf{R}_\gamma^{-1}(\mathbf{t})\mathbf{A}_\gamma^T]^{-1}\mathbf{q}_\gamma = \mathbf{R}_\gamma^{-1}(\mathbf{t})\mathbf{A}_\gamma^T \mathbf{C}_\gamma^{-1}(\mathbf{t})\mathbf{q}_\gamma. \tag{3.2}$$

Itt azonban a $\mathbf{C}_\gamma(\mathbf{t})$ mátrixok már csak 2×2, illetve 3×3-as méretűek, így inverzeiket már explicit módon meg tudjuk adni.

A nagy szerkezet és a kis részszerkezetek közötti kapcsolatot az

$$\mathbf{x}_\gamma = [\mathbf{x}]_\gamma, \quad \gamma = 1, 2, \ldots, M \tag{3.3}$$

egyenlőségek megkövetelésével hozzuk létre. Így hozzuk létre a $\mathbf{q}_\gamma$, $\gamma = 1, 2, \ldots, M$ ún. fiktív erőket a következő képlet alapján:

$$\mathbf{q}_\gamma = \mathbf{C}_\gamma[\mathbf{x}]_\gamma, \quad \gamma = 1, 2, \ldots, M, \tag{3.4}$$

ahol $[\mathbf{x}]_\gamma$ jelöli a megfelelő vektor γ-adik elmozduló csomóponthoz tartozó részvektorát. A $\mathbf{q}_\gamma$-t a további iterációk során mi konstansnak tartjuk és így az $\mathbf{y}_\gamma(\mathbf{t})$ közelítését kapjuk, mint azt [2] és [6] javasolták.

A rúderőkre vonatkozóan a következő lemma bizonyítható [5].

3.1. LEMMA. A $\mathbf{t}=\mathbf{t}^\circ$ esetben a tényleges rúderők a részszerkezetekben számított értékekből a következőképpen kaphatók meg:

$$y^i(\mathbf{t}^\circ) = \begin{cases} y^i_\gamma(\mathbf{t}^\circ), & \text{ha az } i\text{-edik rúd csak a } \gamma\text{-adik csomóponti rendszerben szerepel,} \\ y^i_{\gamma_1}(\mathbf{t}^\circ) + y^i_{\gamma_1}(\mathbf{t}^\circ), & \text{ha az } i\text{-edik rúd a } \gamma_1 \text{ és } \gamma_2 \text{ indexű csomóponti rendszerben is szerepel.} \end{cases} \tag{3.5}$$

(Itt az i index a megfelelő vektor i-edik rúdra vonatkozó részvektorát jelöli).

A fenti lemma adja az ötletet, hogy $\mathbf{t} \neq \mathbf{t}^\circ$ esetben hogyan approximáljuk a rúderőket. Az $\hat{y}^i(\mathbf{t})$-t a (3.5) képlettel definiáljuk $\mathbf{t}^\circ = \mathbf{t}$ helyettesítéssel.

A továbbiakban két megjegyzést teszünk a modellre vonatkozóan.

i) A számítási tapasztalatok alapján a konstansnak választott $\mathbf{q}_\gamma$, $\gamma = 1, 2, \ldots, M$ fiktív erők lényegesen függenek a $\mathbf{t}^\circ$ választásoktól. Azaz a $\mathbf{t}^\circ$-tól való kis elmozdulás esetén is a $\mathbf{q}_\gamma$ lényegesen változott és ez lényegesen befolyásolta a (3.5) módosított képlete alapján adódó közelítés minőségét.

ii) Két elmozduló csomópontot összekötő rúd esetén a rúderő közelítése két komponensből tevődik össze. Számítási tapasztalatok azt mutatják, hogy gyakran ezek a komponensek ellenkező előjelűek, és a részszerkezetbeli értékek abszolútértékei is lényegesen különböznek a tényleges rúderők abszolútértékeitől, azaz a részszerkezetek viselkedése lényegesen eltérhet a megfelelő rúdcsoportok nagyszerkezetbeli viselkedésétől, még a $\mathbf{t}=\mathbf{t}^\circ$ esetén is.

Ezek az észrevételek azt mutatják, hogy annak érdekében, hogy a részrendszerek viselkedése hasonló legyen az eredeti viselkedéshez, a modell további finomítására van szükség. Ezt részletezzük a következő fejezetben.

4. A módosított felbontási elv

Az előző fejezetben ismertettük az általunk készített modell alapját képező felbontást. A részszerkezetek most is a korábbiak lesznek azzal a különbséggel, hogy a rudak szabad végpontjait most nem rögzítjük. A részszerkezeteket most úgy akarjuk ún. fiktív erőkkel terhelni, hogy a rúderők az adott $\mathbf{t}^\circ$ keresztmetszet mellett megegyezzenek az eredeti rúderőkkel. A korábbi jelöléseket figyelembe véve:

$$\mathbf{y}_\gamma = [\mathbf{y}]_\gamma \tag{4.1}$$

teljesülését kívánjuk biztosítani. A következő levezetéseket egy rögzített $\mathbf{t}^\circ$ keresztmetszetre végezzük, és később terjesztjük ki tetszőleges $\mathbf{t}$-re.

Az $\mathbf{R}(\mathbf{t})$ mátrix blokkos struktúrájából következik, hogy az $\mathbf{R}(\mathbf{t})\,\mathbf{y}$ szorzatban részvektorként éppen az $\mathbf{R}_\gamma(\mathbf{t})\,[\mathbf{y}]_\gamma$ szorzat is szerepel. Így a (2.1) megfelelő részéből kapjuk, hogy

$$\mathbf{R}_\gamma(\mathbf{t})[\mathbf{y}]_\gamma - [\mathbf{A}^T\mathbf{x}]_\gamma = \mathbf{0}, \tag{4.2}$$

ahol $[\mathbf{A}^T\mathbf{x}]_\gamma$ az $\mathbf{A}^T\mathbf{x}$ szorzatvektor megfelelő részét, komponenseit tartalmazza. Ha $[\mathbf{A}^T]_\gamma$ az $\mathbf{A}^T$ mátrix γ-adik csomóponthoz tartozó soraiból áll, akkor a következő egyenlőség áll fenn:

$$[\mathbf{A}^T\mathbf{x}]_\gamma = [\mathbf{A}^T]_\gamma\mathbf{x}. \tag{4.3}$$

Ezt felhasználva (4.2)-ben azt kapjuk, hogy

$$[\mathbf{y}]_\gamma = \mathbf{R}_\gamma^{-1}(\mathbf{t})[\mathbf{A}^T]_\gamma\mathbf{x}. \tag{4.4}$$

A további levezetés alapja az az észrevétel, hogy $\mathbf{A}_\gamma^T$ részmátrixként szerepel az $[\mathbf{A}^T]_\gamma$ mátrixban. Ezért bontsuk fel az $[\mathbf{A}^T]_\gamma$ oszlopait a következőképpen:

$$[\mathbf{A}^T]_\gamma = [\mathbf{A}_\gamma^T, \mathbf{Q}_\gamma^T]$$

majd pedig helyettesítsük be a (4.4) kifejezésbe. Így nyerjük, hogy

$$[\mathbf{y}]_\gamma = \mathbf{R}_\gamma^{-1}(\mathbf{t})(\mathbf{A}_\gamma^T[\mathbf{x}]_\gamma + \mathbf{Q}_\gamma^T[\mathbf{x}]_{\bar\gamma}), \tag{4.5}$$

ahol $[\mathbf{x}]_\gamma$ a γ-adik csomópont elmozdulásvektora, míg az $[\mathbf{x}]_{\bar\gamma}$ a többi csomópontok elmozdulásait tartalmazza. A (3.1)-ből a részrendszerbeli erővektorra a következő kifejezést kapjuk:

$$\mathbf{y}_\gamma = \mathbf{R}_\gamma^{-1}(\mathbf{t})\mathbf{A}_\gamma^T\mathbf{x}_\gamma. \tag{4.6}$$

Felhasználva, hogy $\mathbf{A}_\gamma\mathbf{y}_\gamma = \mathbf{q}_\gamma$ a fiktív erőkre a

$$\mathbf{q}_\gamma = \mathbf{A}_\gamma\mathbf{R}_\gamma^{-1}(\mathbf{t})\mathbf{A}_\gamma^T\mathbf{x}_\gamma \tag{4.7}$$

előállítást kapjuk. Emlékeztetünk arra, hogy a célkitűzésünk az $\mathbf{y}_\gamma = [\mathbf{y}]_\gamma$ egyenlőség teljesítése, azaz a (4.7) összefüggés helyett olyan relációt keresünk, amely (4.1) teljesülését biztosítja. A (4.5) és (4.6) relációból adódik, hogy a kívánt egyenlőség akkor és csak akkor teljesül, ha

$$\mathbf{A}_\gamma^T\mathbf{x}_\gamma = \mathbf{A}_\gamma^T[\mathbf{x}]_\gamma + \mathbf{Q}_\gamma^T[\mathbf{x}]_{\bar\gamma} \tag{4.8}$$

fennáll.

Könnyen látható, hogy (4.8) $\mathbf{x}_\gamma$-ban túlhatározott lineáris egyenletrendszer, azaz csak speciális esetekben oldható meg. Ebből az következik, hogy a *Halmos—Rapcsák*

modellben használt részszerkezetek esetén csak a csomóponti fiktív terhelésekkel az $[\mathbf{y}]_\gamma = \mathbf{y}_\gamma$ egyenlőséget nem tudjuk biztosítani.

A fenti egyenlőség elérésének érdekében a

$$\mathbf{x}_\gamma = [\mathbf{x}]_\gamma \tag{4.9}$$

feltevéssel élünk (4.6)-ban, azaz ez azt jelenti, hogy a csomópont elmozdulása a részszerkezetben megegyezik az eredeti szerkezetbeli elmozdulásával. Így a kétféle rúderő közötti kapcsolat a következőképpen néz ki a (4.5) és a módosított (4.6) felhasználásával:

$$[\mathbf{y}]_\gamma = \mathbf{y}_\gamma + \mathbf{R}_\gamma^{-1}(\mathbf{t})\mathbf{Q}_\gamma^T[\mathbf{x}]_{\bar{\gamma}} \tag{4.10}$$

Így látható, hogy az eredeti modellben szereplő rúderőkhöz az $\mathbf{R}_\gamma^{-1}(\mathbf{t})\,\mathbf{Q}_\gamma^T[\mathbf{x}]_{\bar{\gamma}}$ értéket kell hozzáadnunk, hogy a tényleges rúderőket megkapjuk. Ezt a hozzáadandó kifejezést könnyű interpretálni. A $\mathbf{Q}_\gamma^T$ mátrix definíciójából adódik, hogy ezek az erők akkor keletkeznek, ha a γ-adik csomópontot rögzítjük és a rúdvégpontokat a szomszédos csomópontokra a (2.1) megoldásakor kapott értéknek megfelelően elmozdítjuk.

A részszerkezetre nézve ez azt jelenti, hogy a csomópontra ható statikus terhelés mellett egy kinematikus terhelést

$$\mathbf{f}_\gamma = \mathbf{Q}_\gamma^T[\mathbf{x}]_{\bar{\gamma}} \tag{4.11}$$

is be kell vezetni. Ennek következtében a részszerkezetre vonatkozó alaprendszert is módosítani kell, mert csak így tudjuk biztosítani a kívánt (4.1) összefüggést:

$$\begin{aligned} \mathbf{R}_\gamma(\mathbf{t})\mathbf{y}_\gamma - \mathbf{A}_\gamma^T\mathbf{x}_\gamma &= \mathbf{f}_\gamma \\ \mathbf{A}_\gamma\mathbf{y}_\gamma &= \mathbf{q}_\gamma \end{aligned} \qquad \gamma = 1, 2, \ldots, M. \tag{4.12}$$

Ebből kapjuk $\mathbf{t}=\mathbf{t}^\circ$ és $\mathbf{x}_\gamma=[\mathbf{x}]_\gamma$ mellett, hogy

$$\mathbf{y}_\gamma(\mathbf{t}^\circ) = \mathbf{R}_\gamma^{-1}(\mathbf{t}^\circ)\mathbf{A}_\gamma^T[\mathbf{x}]_\gamma + \mathbf{R}_\gamma^{-1}(\mathbf{t}^\circ)\mathbf{f}_\gamma \tag{4.13}$$

és a $\mathbf{q}_\gamma$ csomóponti terhelésre

$$\begin{aligned} \mathbf{q}_\gamma = \mathbf{A}_\gamma\mathbf{y}_\gamma(\mathbf{t}^\circ) &= \mathbf{A}_\gamma\mathbf{R}_\gamma^{-1}(\mathbf{t}^\circ)\mathbf{A}^T[\mathbf{x}]_\gamma + \mathbf{A}_\gamma\mathbf{R}_\gamma^{-1}(\mathbf{t}^\circ)\mathbf{f}_\gamma = \\ &= \mathbf{C}_\gamma(\mathbf{t}^\circ)[\mathbf{x}]_\gamma + \mathbf{A}_\gamma\mathbf{R}_\gamma^{-1}(\mathbf{t})\mathbf{f}_\gamma \end{aligned} \tag{4.14}$$

kifejezést kapjuk. Ezt felhasználva az $[\mathbf{x}]_\gamma$ kifejezésére, majd pedig behelyettesítve (4.13)-ba a γ-adik csomópont rúderőire a következő kifejezést kapjuk:

$$\mathbf{y}_\gamma(\mathbf{t}^\circ) = \mathbf{R}_\gamma^{-1}(\mathbf{t}^\circ)\mathbf{A}^T[\mathbf{C}_\gamma^{-1}(\mathbf{t}^\circ)_\gamma - \mathbf{C}_\gamma^{-1}(\mathbf{t}^\circ)\mathbf{A}_\gamma\mathbf{R}_\gamma^{-1}(\mathbf{t}^\circ)\mathbf{f}_\gamma] + \mathbf{R}_\gamma^{-1}(\mathbf{t}^\circ)\mathbf{f}_\gamma. \tag{4.15}$$

Így azt kapjuk, hogy a csomóponti terhelést a (4.14) kifejezés alapján számoljuk, akkor $\mathbf{t}=\mathbf{t}^\circ$-ra fennáll a kívánt összefüggés:

$$[\mathbf{y}(\mathbf{t}^\circ)]_\gamma = \mathbf{y}_\gamma(\mathbf{t}^\circ)$$

azaz a részszerkezet rúdjaiban ébredő erő megegyezik a nagy, eredeti szerkezetbeli értékkel. Az eredeti modellel analóg módon a (4.15) reláció lesz az explicit közelítése a rúderőknek $\mathbf{t}\neq\mathbf{t}^\circ$ esetben:

$$\hat{\mathbf{y}}_\gamma(\mathbf{t}) = \mathbf{R}_\gamma^{-1}(\mathbf{t})\mathbf{A}_\gamma^T[\mathbf{C}_\gamma^{-1}(\mathbf{t})\mathbf{q}_\gamma - \mathbf{C}_\gamma^{-1}(\mathbf{t})\mathbf{A}_\gamma\mathbf{R}_\gamma^{-1}(\mathbf{t})\mathbf{f}_\gamma] + \mathbf{R}_\gamma^{-1}(\mathbf{t})\mathbf{f}_\gamma. \tag{4.16}$$

Így explicit közelítést kaptunk a részszerkezet rúderőire, amikor $\mathbf{q}_\gamma$-t és $\mathbf{f}_\gamma$-t a megfelelő kifejezés alapján számoljuk és konstansnak tartjuk. A csomópontbeli $\mathbf{q}_\gamma$ erőkre a következő tulajdonságot lehet bizonyítani

4.1. LEMMA: Legyenek a $\mathbf{q}_\gamma$ csomóponti erők a (4.14) alapján számolva, ahol $\mathbf{f}_\gamma$-t a (4.11) képlet adja meg, akkor a következők állnak fenn:

i) az olyan csomópontokban, amelyek minden irányban szabadon el tudnak mozdulni a $\mathbf{q}_\gamma$ a $\mathbf{t}^\circ$ választásától független, és az eredeti szerkezet γ-adik csomópontjában ható külső terheléssel egyezik meg, azaz ha $[\mathbf{q}]_\gamma$ a $\mathbf{q}$ külső terhelésvektor γ-adik csomóponthoz tartozó részvektorát jelöli, akkor

$$\mathbf{q}_\gamma = [\mathbf{q}]_\gamma$$

ii) az olyan csomópontban, amely valamely irány(ok)-ban rögzítve van(ak), akkor a $\mathbf{q}_\gamma$ megfelelő komponensei az ezen irányban ébredő reakciós erővel (erőkkel) egyezik (egyeznek) meg, míg a szabad irányoknak megfelelő komponensek az i) szerinti értékeket veszik fel.

Bizonyítás: Ezen tulajdonságok bizonyítására az $\mathbf{A}_\gamma$ mátrix speciális tulajdonságát használjuk ki. Legyen $\mathbf{a}_\gamma^i$ az $\mathbf{A}_\gamma$ egy tetszőleges sora, ekkor fennáll, hogy

$$\mathbf{a}_\gamma^i \mathbf{y}_\gamma = q_\gamma^i. \tag{4.17}$$

Ez az egyenlet azt fejezi ki, hogy egy adott irányban a rúderők eredője a $\mathbf{q}_\gamma$ megfelelő komponensével van egyensúlyban. A következő eseteket kell megkülönböztetnünk:

a) A γ-adik csomópont ebben az irányban szabadon elmozdulhat. Ekkor a (2.1) alaprendszerben létezik egy ennek az iránynak megfelelő erőegyensúly összefüggés, azaz létezik $\mathbf{a}^j$ sora $\mathbf{A}$-nak, amelyben $\mathbf{a}_\gamma^i$ részvektorként szerepel. Ebből következik, hogy

$$\mathbf{a}^j \mathbf{y} = \mathbf{a}_\gamma^i [\mathbf{y}]_\gamma + \mathbf{a}_{\bar{\gamma}}^i [\mathbf{y}]_{\bar{\gamma}} = [\mathbf{q}]^j, \tag{4.18}$$

ahol $[\mathbf{y}]_{\bar{\gamma}}$ az összes a γ-adik csomóponthoz nem tartozó rudak erőkomponenseit tartalmazza, míg az $\mathbf{a}^j$ megfelelő felbontása: $\mathbf{a}^j = (\mathbf{a}_\gamma^i, \mathbf{a}_{\bar{\gamma}}^i)$, és $[\mathbf{q}]^j$ jelöli a $\mathbf{q}$ külső terhelésvektor γ-adik csomópont megfelelő irányának megfelelő komponensét. Mivel a γ-adik részszerkezet az összes a γ-adik csomópontba befutó rudat tartalmazza, így az $\mathbf{a}_{\bar{\gamma}}^i$ nullvektor lesz, ami azt jelenti, hogy az $\mathbf{A}_\gamma$ mátrix képzésekor az $\mathbf{a}^j$ sorból elhagyott elemek mind nullák. Mivel fennáll, hogy $\mathbf{y}_\gamma(\mathbf{t}^\circ) = [\mathbf{y}(\mathbf{t}^\circ)]_\gamma$ a (4.17) és (4.18)-ból következik, hogy a $\mathbf{t}^\circ$ választásától függetlenül

$$\mathbf{q}_\gamma = [\mathbf{q}]_\gamma.$$

b) Ha a γ-adik csomópont az adott irányban rögzített a (2.1)-ben ennek az iránynak megfelelően nincs erőegyensúly összefüggés (azaz nincs megfelelő sor $\mathbf{A}$-ban), mivel a rögzítés, ameddig a terhelést a reakcióerőn keresztül elbírja, az erőegyensúlyt mindig biztosítja. Mivel a γ-adik csomóponti részszerkezetben ennek az iránynak megfelelően is ugyanazok az erők lépnek fel, mint az eredeti nagy szerkezetben, emiatt a (4.17) felhasználásával a részszerkezet erőegyensúlyának következtében $\mathbf{q}_\gamma^i$-nek meg kell egyeznie a reakcióerővel. Ezzel a lemmát bizonyítottuk is.

Összegezve leszögezhetjük, hogy a $\mathbf{q}_\gamma$ konstansnak feltételezése ebben a modellben jogos, kivéve a rögzített csomópontokat, ahol reakcióerő is fellép. Még meg kell vizsgálnunk, hogy az $\mathbf{f}_\gamma = \mathbf{Q}_\gamma^T [\mathbf{x}]_{\bar{\gamma}}$ kinematikus terhelés hogyan függ $\mathbf{t}^\circ$ változásától.

Lineárisan elasztikus rúdszerkezetekre $\alpha>0$ esetén fennáll, hogy

$$\mathbf{R}^{-1}(\alpha\mathbf{t}) = \alpha\mathbf{R}^{-1}(\mathbf{t}) \tag{4.19}$$

és $\mathbf{C}$ definíciójából közvetlenül következik, hogy

$$\mathbf{C}^{-1}(\alpha\mathbf{t}) = \frac{1}{\alpha}\mathbf{C}^{-1}(\mathbf{t}). \tag{4.20}$$

Mivel $\mathbf{Cx}=\mathbf{q}$ minden $\mathbf{t}$ keresztmetszetvektor esetén, így

$$\mathbf{x}(\alpha\mathbf{t}) = \frac{1}{\alpha}\mathbf{x}(\mathbf{t}) \tag{4.21}$$

$$\mathbf{y}(\alpha\mathbf{t}) = \mathbf{y}(\mathbf{t}). \tag{4.22}$$

A (4.22) tulajdonságot szeretnénk elérni a (4.16) által előálló $\hat{\mathbf{y}}_\gamma(\mathbf{t})$ közelítő rúderővektorokra. A (4.21) egyenletet figyelembe véve az eredeti szerkezet csomópontjaira szorzófaktorokat, súlyokat vezetünk be:

$$\alpha_\beta(\mathbf{t}) = \frac{\sum\limits_{n\in I_\beta} t_i^\circ}{\sum\limits_{i\in I_\beta} t_i} \qquad \beta = 1, 2, \ldots, M, \tag{4.23}$$

ahol I_β a β-adik csomópontba befutó rudak indexeinek halmaza. Jelöljük $[\hat{\mathbf{x}}(\mathbf{t})]_{\bar\gamma}$-sal a $[\mathbf{x}(\mathbf{t}^\circ)]_{\bar\gamma}$ súlyozottját, ami azt jelenti, hogy az $[\mathbf{x}(\mathbf{t}^\circ)]_{\bar\gamma}$ minden komponensét a megfelelő csomópont szorzófaktorával megszorozzuk. Jelölje ezek után

$$\hat{\mathbf{f}}_\gamma(\mathbf{t}) = \mathbf{Q}_{\bar\gamma}^T[\hat{\mathbf{x}}(\mathbf{t})]_{\bar\gamma}.$$

A következő állítást bizonyíthatjuk:

4.2. LEMMA: Ha egy lineárisan elasztikus rúdszerkezet esetén a (4.16) kifejezésben az $\mathbf{f}_\gamma$-t $\hat{\mathbf{f}}_\gamma$-val helyettesítjük, akkor fennáll, hogy

$$\hat{\mathbf{y}}_\gamma(\alpha\mathbf{t}^\circ) = \hat{\mathbf{y}}_\gamma(\mathbf{t}^\circ), \quad \alpha > 0 \tag{4.24}$$

és

$$\hat{\mathbf{y}}_\gamma(\mathbf{t}^\circ) = [\mathbf{y}(\mathbf{t}^\circ)]_\gamma. \tag{4.25}$$

Bizonyítás: A (4.25) egyenlőség a (4.15) és a (4.16) közvetlen következménye, mivel $\hat{\mathbf{f}}_\gamma(\mathbf{t}^\circ)=\mathbf{f}_\gamma$ és $\hat{\mathbf{y}}_\gamma(\mathbf{t}^\circ)=\mathbf{y}_\gamma(\mathbf{t}^\circ)$ fennáll. Az első egyenlőség bizonyítása végett a (4.16)-ban az $\mathbf{f}_\gamma$ helyére helyettesítsük be $\hat{\mathbf{f}}_\gamma$-t.

A (4.16) így kapott módosított alakjában (4.19), (4.20) felhasználásával kapjuk, hogy

$$\hat{\mathbf{y}}_\gamma(\alpha\mathbf{t}^\circ) = \alpha\mathbf{R}_{\bar\gamma}^{-1}(\mathbf{t}^\circ)\mathbf{A}_{\bar\gamma}^T\left[\frac{1}{\alpha}\mathbf{C}_{\bar\gamma}^{-1}(\mathbf{t}^\circ)\mathbf{q}_{\bar\gamma} - \frac{1}{\alpha}\mathbf{C}_{\bar\gamma}^{-1}(\mathbf{t}^\circ)\mathbf{A}_{\bar\gamma}(\alpha\mathbf{R}_{\bar\gamma}^{-1}(\mathbf{t}^\circ))\hat{\mathbf{f}}_{\bar\gamma}(\alpha\mathbf{t}^\circ)\right] +$$

$$+\alpha\mathbf{R}_{\bar\gamma}^{-1}(\mathbf{t}^\circ)\hat{\mathbf{f}}_{\bar\gamma}(\alpha\mathbf{t}^\circ).$$

A (4.23)-ból azonnal adódik:

$$\hat{\mathbf{f}}_\gamma(\alpha\mathbf{t}^\circ) = \frac{1}{\alpha}\hat{\mathbf{f}}_\gamma(\mathbf{t}^\circ) = \frac{1}{\alpha}\mathbf{f}_\gamma.$$

Ezt behelyettesítve a fenti egyenletbe, a kívánt egyenlőséget kapjuk. Így ezt a lemmát is bizonyítottuk.

A fentiekben leírt approximáció alkalmazása a következő lépésekben történik.

a) Egy kiválasztott $\mathbf{t}^\circ$ keresztmetszetvektor mellett a (2.1) alapegyenletrendszert megoldjuk, ahonnan a $[\mathbf{x}(\mathbf{t}^\circ)]_\gamma$ és $[\mathbf{x}(\mathbf{t}^\circ)]_{\bar{\gamma}}$ csomóponti vektorokat kapjuk. A (4.11) alapján $\mathbf{f}_\gamma$, (4.14) alapján pedig a $\mathbf{q}_\gamma$ értékeket meghatározzuk.

b) A (4.16) $\hat{\mathbf{f}}_\gamma$-val módosított kifejezése alapján $\mathbf{t} \neq \mathbf{t}^\circ$ esetben a rúderővektorok közelítő értékeit nyerjük.

Az új modellre vonatkozóan néhány megjegyzést teszünk.

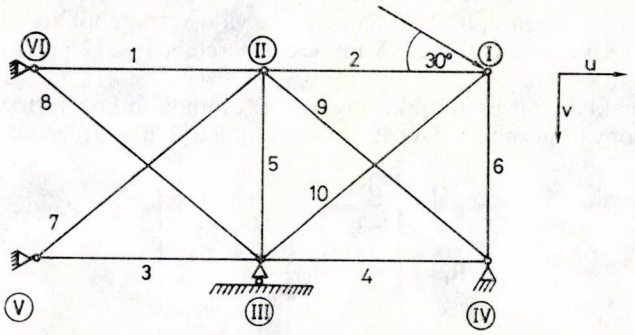
1) Az $\hat{\mathbf{f}}_\gamma = \mathbf{Q}_\gamma^T [\mathbf{x}(\mathbf{t})]_{\bar{\gamma}}$ kinematikai terhelés lineárisan elasztikus szerkezetekre való bevezetése után ugyanezt a súlyozást használtuk más rúdszerkezetek esetén is. A teszteredmények tapasztalatai alapján ez a súlyozás ilyen rúdszerkezetek esetén is előnyös.

2) Az olyan rudak, amelyek két elmozduló csomópontot kötnek össze, két részszerkezetben is szerepelnek, így a rúderővektorra két közelítőértéket is kapunk (ezek az értékek $\mathbf{t}=\mathbf{t}^\circ$ esetén megegyeznek!) Amennyiben ezen két érték között lényeges eltérés mutatkozik az approximáció a) lépésének megismétlése szükséges az adott $\mathbf{t}$ keresztmetszet $\mathbf{t}^\circ$-nak választásával.

3) A modell részszerkezeteinek tulajdonságait vizsgálva észrevehetjük, hogy a csomópontok elmozdulásai megegyeznek az eredeti szerkezetbeli értékkel. A kinematikus terhelés definíciója által elértük, hogy a szomszédos csomópontok elmozdulásai is megegyeznek az eredeti értékekkel. Ezzel elértük, hogy a részszerkezetek mechanikai tulajdonságai megegyeznek az egész szerkezetbeli viselkedéssel, ami a rúderők megegyezésében is megmutatkozik. Így mód nyílik egy nagyméretű szerkezet pl. két részre bontására, majd pedig a két „fél" szerkezet fenti modell által való méretezésére egymástól teljesen függetlenül.

5. Példa és teszteredmények

A modell fő lépéseit egy 10 rúdból álló rácsos tartó példáján fogjuk megmutatni. A tartó a következőképpen néz ki:



1. ábra

Amint az 1. ábrán is látjuk a tartó 3 elmozduló csomópontot tartalmaz (I., II., III.), ahol a III. csak u irányban tud elmozdulni. A (2.1) alaprendszer felírásához szükséges az **R**, **A** mátrixok, valamint a külső terhelés **q** vektorának ismerete. Ezen rácsos tartó **R** mátrixa egy diagonálmátrix, amely a következő alakú:

$$\mathbf{R} = \begin{pmatrix} \frac{1}{k_1 t_1} & & & 0 \\ & \frac{1}{k_2 t_2} & & \\ & & \ddots & \\ 0 & & & \frac{1}{k_{10} t_{10}} \end{pmatrix}$$

ahol k_i, $i=1, 2, \ldots, 10$ az E rugalmassági együttható és a rúd hosszának hányadosa, azaz $k_i = \frac{E}{l_i}$ $i=1, 2, \ldots, 10$, és t_i az i-edik rúd keresztmetszete.

Az **A** mátrix felírásához először a tartó minden rúdjához egy irányítást kell rendelnünk. Ezt az irányítást a következőképpen adjuk meg. A rúd kezdőpontja a kisebb indexű, végpontja a nagyobb indexű csomópont. Mivel az I., II. csomópontok mindkét irányba, míg a III. csomópont csak v irányba tud elmozdulni, így az **A** mátrix 5 sorból fog állni.

	1	2	3	4	5	6	7	8	9	10
I.		a_{12}				a_{16}				$a_{1,10}$
		b_{12}				b_{16}				$b_{1,10}$
II.	a_{21}	a_{22}			a_{25}		a_{27}		a_{29}	
	b_{21}	b_{22}			b_{25}		b_{27}		b_{29}	
III.			a_{33}	a_{34}	a_{35}			a_{38}		$a_{3,10}$

Itt a_{ij} jelenti a j-edik rúd iránykoszinuszát az u irányhoz viszonyítva és b_{ij} a v irányhoz viszonyítva. A **q** vektor ennek megfelelően 5 komponensből fog állni, ahol az első két komponens a $\mathbf{q}_I$ u, ill. v tengelyekre való vetítéséből adódik, míg a megmaradó három komponens 0 lesz. A fentiek ismeretében az (2.1) alaprendszer felírható a fenti tartó esetén.

A továbbiakban megmutatjuk, hogy az I. csomóponthoz tartozó részrendszer részmátrixai hogyan néznek ki. Az I. csomópontba a 2, 6 és 10 rudak futnak be, így

$$\mathbf{R}_I = \begin{pmatrix} \frac{1}{k_2 t_2} & & 0 \\ & \frac{1}{k_6 t_6} & \\ 0 & & \frac{1}{k_{10} t_{10}} \end{pmatrix}$$

és az $\mathbf{A}_I$ 2×3-as mátrix

$$\mathbf{A}_I = \begin{pmatrix} a_{12} & a_{16} & a_{1,10} \\ b_{12} & b_{16} & b_{1,10} \end{pmatrix}$$

alakúak. A $\mathbf{C}_I$ mátrix a fenti mátrixok megfelelő szorzatából adódik, azaz:

$$\mathbf{C}_I = \mathbf{A}_I \mathbf{R}_I^{-1} \mathbf{A}_I^T = \begin{pmatrix} a_{12} & a_{16} & a_{1,10} \\ b_{12} & b_{16} & b_{1,10} \end{pmatrix} \begin{pmatrix} k_2 t_2 & 0 & 0 \\ 0 & k_6 t_6 & 0 \\ 0 & 0 & k_{10} t_{10} \end{pmatrix} \begin{pmatrix} a_{12} & b_{12} \\ a_{16} & b_{16} \\ a_{1,10} & b_{1,10} \end{pmatrix}.$$

A $\mathbf{C}_I$ mátrix látható, hogy szimmetrikus, 2×2 mátrix, így a következő alakú:

$$\mathbf{C}_I = \begin{pmatrix} c_1 & c_2 \\ c_2 & c_3 \end{pmatrix}$$

és így az inverz expliciten megadható:

$$C_I^{-1} = \frac{1}{C_1 C_3 - C_2^2} \begin{pmatrix} C_3 & -C_2 \\ -C_2 & C_1 \end{pmatrix}.$$

Az eredeti szerkezet alaprendszerének $\mathbf{t}=\mathbf{t}^\circ$ esetben való megoldásakor az $\mathbf{x}$ megoldásvektor 5 komponenséből az I. csomóponthoz tartozó két komponens alkotja az $\mathbf{x}_I(\mathbf{t}^\circ)$-at, míg a többi három az $\mathbf{x}_I(\mathbf{t}^\circ)$-t. Az $\hat{\mathbf{f}}_I$ meghatározásához a $\mathbf{Q}_I^T$ szükséges még. Tekintsük a (10×5)-ös $\mathbf{A}^T$ mátrixot, majd pedig válaszzuk ki a 2,6, 10 rudakhoz tartozó három sorát, ekkor a következő mátrix áll elő:

$$\left(\begin{array}{cc:ccc} a_{12} & b_{12} & a_{22} & b_{22} & 0 \\ a_{16} & b_{16} & 0 & 0 & 0 \\ a_{1,10} & b_{1,10} & 0 & 0 & a_{3,10} \end{array}\right).$$

A szaggatott vonal osztja ezt szét a már korábban felírt $\mathbf{A}_I$ mátrix $\mathbf{A}_I^T$-jára valamint a $\mathbf{Q}_I^T$ mátrixra. Ekkor már $\mathbf{f}_I$ megkapható a következő szorzat eredményeképpen:

$$\mathbf{f}_I = \begin{pmatrix} a_{22} & b_{22} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & a_{3,10} \end{pmatrix} \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} = \mathbf{Q}_I^T \mathbf{x}_I.$$

A $\hat{\mathbf{f}}_I$ meghatározásához még szükségesek a megfelelő súlyozófaktorok meghatározása. A definíció alapján látható, hogy a II. és a III. csomópontra egy tetszőleges $\mathbf{t}$ vektor esetén a következőképpen néznek ki:

$$\alpha_{II}(\mathbf{t}) = \frac{t_1^\circ + t_2^\circ + t_5^\circ + t_7^\circ + t_9^\circ}{t_1 + t_2 + t_5 + t_7 + t_9}$$

$$\alpha_{III}(\mathbf{t}) = \frac{t_3^\circ + t_4^\circ + t_5^\circ + t_8^\circ + t_{10}^\circ}{t_3 + t_4 + t_5 + t_8 + t_{10}}.$$

Ennek megfelelően az $\hat{\mathbf{f}}_{\mathrm{I}}(\mathbf{t})$ a következőképpen áll elő:

$$\hat{\mathbf{f}}_{\mathrm{I}}(\mathbf{t}) = \begin{pmatrix} a_{22} & b_{22} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & a_{3,10} \end{pmatrix} \begin{pmatrix} x_3 & \alpha_{\mathrm{II}}(\mathbf{t}) \\ x_4 & \alpha_{\mathrm{II}}(\mathbf{t}) \\ x_5 & \alpha_{\mathrm{III}}(\mathbf{t}) \end{pmatrix}.$$

Ezek az eredmények felhasználásával a (4.14) és a (4.16) képletek alapján az $\mathbf{y}_{\mathrm{I}}=(y_2, y_6, y_{10})$ részvektor approximációját kapjuk.

A fenti rácsos tartó esetén mindkét approximációval számolásokat végeztünk. Kiindultunk a $\mathbf{t}^{\circ}$, $\mathbf{t}_i^{\circ}=1$, $i=1, 2, \ldots 10$ vektorból és különböző $\mathbf{t}$ vektorok esetén számoltuk a tényleges rúderőket a *Halmos—Rapcsák-féle modell* által meghatározott, valamint az új felbontás által meghatározott rúderőket. A teszteredmények azt mutatták, hogy az új modellel számolt rúderők kevésbé térnek el a ténylegestől, mint *Halmos—Rapcsák-féle eredeti modell* esetén. Így az optimalizálás során konstansnak választott $\mathbf{x}(\mathbf{t}^{\circ})$ vektort ritkábban kellett újra meghatározni az eredeti szerkezetre vonatkozó (2.1) rendszer megoldásával.

A következő táblázatokban ismertetjük a számítási eredményeket. Minden táblázat egy adott keresztmetszetvektorra vonatkozik, és felsoroljuk a rudakban keletkező tényleges erőt és feszültséget, valamint a két modell (H—R, valamint az ÚJ) által szolgáltatott közelítéseket.

	1	2	3	4	5	6	7	8	9	10
$\mathring{\mathbf{t}}$	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
$\mathbf{y}(\mathring{\mathbf{t}})$	−296	−615	−99	99	0	679	−196	−65	196	−308
σ	296	615	99	99	0	679	196	65	196	308
$\mathbf{y}(\mathring{\mathbf{t}})$ H—R	−296	−615	−99	99	0	679	−196	−65	196	−308
$\mathbf{y}(\mathring{\mathbf{t}})$ ÚJ	−296	−615	−99	99	0	679	−196	−65	196	−308

$\mathbf{t}$	0,95	1,1	0,95	0,95	0,95	1,45	0,95	0,95	0,95	0,95
$\mathbf{y}(\mathbf{t})$	−291	−605	−102	102	0	685	−193	−68	193	−319
σ	306	550	107	107	0	473	203	71	203	336
$\mathbf{y}(\mathbf{t})$ H—R	−281	−586	−98	98	0	678	−186	−66	186	−306
$\mathbf{y}(\mathbf{t})$ ÚJ	−265	−606	−80	80	0	685	−175	−63	175	−319

t	1,7	1,5	0,2	0,2	0.2	1,0	0,2	0,2	0,2	0,2
y(t)	−714	−805	−24	24	0	543	−56	−16	56	−75
σ	420	537	119	119	0	543	278	79	278	371
y(t) H—R	−453	−759	−98	98	0	501	−35	−65	35	−3
y(t) ÚJ	−662	−844	−98	98	0	515	−58	−55	58	−17

t	1,44	1,34	0,2	0,2	0,2	0,9	0,2	0,2	0,2	0,2
y(t)	−694	−797	−26	26	0	548	−64	−18	64	−84
σ	481	595	134	134	0	609	319	89	319	418
y(t) H—R	−437	−745	−99	99	0	506	−40	−65	40	−105
y(t) ÚJ	−630	−829	−24	24	0	520	−66	−55	66	−36

t	1,09	1,31	0,2	0,2	0,2	0,93	0,2	0,2	0,2	0,2
y(t)	−656	−786	−31	−31	0	556	−79	−20	79	−98
σ	602	600	157	157	0	599	399	103	399	489
y(t) H—R	−379	−694	−99	99	0	506	−46	−65	46	−14
y(t) ÚJ	−611	−732	−24	24	0	526	−74	−56	74	−45

IRODALOM

[1] BERNAU H., HALMOS, E., "Dimensioning of staticallsy indeterminate lightweight structures of complex stress on the basis of minimum-weight conditions" *MTA SZTAKI, Working paper,* MO **21** (1980).

[2] FLEURY, C. and GERADIN, M., "Optimality criteria and mathematical programming in structural weight optimization", *Computer and Structures* **8** (1978) 7—17

[3] GALLANGER, R. M., *Finite Element Analysis,* (Springer-Verlag, Berlin, Heidelberg, New York, 1976.

[4] HALMOS, E. and RAPCSÁK, T., "Minimum weight design of thc statically indeterminate trusses", *Mathematical Programming Study* **9** (1978) 109—119.

[5] HALMOS, E. és RAPCSÁK, T., „Statikailag határozatlan rácsos tartók minimális súlyra történő méretezése" *Alkalmazott Matematikai Lapok* **3** (1977) 171—183.

[6] SANDER, G. and FLEURY, C., "A mixed method in structural optimization", *International Journal for Numerical Methods in Engineering* **13** (1978) 385—404.

[7] SZABÓ, J. és ROLLER, B., *Rúdszerkezetek elmélete és számítása* (Műszaki Könyvkiadó, Budapest, 1971).
[8] VENHAYYA, V. B., "Stuctural optimization a review and some recommendations", *International Journal for Numerical Methods in Engineering* **13** (1978) 203—228.

BERNAU HEINZ
SOÓS ZSOLT
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1111 BUDAPEST, KENDE UTCA 13—17.

HALMOS EMIL
KÖZLEKEDÉSI ÉS TÁVKÖZLÉSI MŰSZAKI FŐISKOLA
9026 GYŐR, SÁGVÁRI E. ÚT 25.

A NEW MODELL FOR THE DETERMINATION OF OPTIMAL TRUSSES

H. BERNAU and E. HALMOS and Zs. SOÓS

For the determination of optimal trusses a decomposition principle is given, by which the investigations can be reduced to the analysis of smaller subtrusses. Furthermore it allows an explicite approximation for the dependence of stresses from cross-section areas.

Alkalmazott Matematikai Lapok 6 (1980) 407—442

MÓDSZEREK ÉS PROGRAMOK RITKA MÁTRIXOKRA

GERGELY JÓZSEF

Budapest

A dolgozatban módszereket és azok programjait ismertetjük 6 lineáris algebrai feladat megoldására, abban az esetben, amikor a feladat mátrixa kevés nem zéró elemet tartalmaz, ritka *(sparse)*. Az első négy módszer, illetve program lineáris egyenletrendszer megoldására szolgál, az ötödik és a hatodik sajátértékeket számol szimmetrikus, nem szimmetrikus ritka, illetve szalagmátrixok esetén. Az első programmal számolhatjuk a mátrix inverzét és determinánsát is.

1. Bevezetés

A számítógépek elterjedése után lehetőség nyílt nagyméretű lineáris rendszerek gépi kezelésére. Ezzel egyidejűleg megnőttek az igények is nagyméretű feladatok megoldására, illetve a korábbi igények reális alapot nyertek. A megoldandó problémák többségében nagyméretű mátrixok segítségével tudjuk megfogalmazni a feladatokat. Az itt megfogalmazott feladatok a lineáris algebra feladatai: lineáris egyenletrendszerek megoldásai, mátrixok invertálása, sajátértékeinek kiszámítása.

Számtalan módszer, program ismeretes lineáris algebrai feladatok megoldására. Lineáris algebrával vagy numerikus módszerekkel foglalkozó tankönyvek külön fejezete vagy fejezetei tárgyalják a lineáris egyenletrendszerek megoldásának, a mátrixok invertálásának a módszereit (lásd pl. [1]). A számítóközpontok programkönyvtárában megtalálhatók a lineáris algebrai feladatok megoldására szolgáló programok.

Nagyméretű lineáris feladatok gépi megoldásához a számítógépekben nagyméretű operatív memóriára van szükségünk. Ha az operatív memória nem elég a mátrix tárolsára, akkor a feladat megoldását valamilyen külső tároló segítségével végezhetjük el. Ekkor a megoldási módszerek programjai bonyolultabbak lesznek és a memóriák közötti adatátvitelek nagyon megnövelik a számítási időt (lásd [2]).

Feladataink többségében a nagyméretű mátrixok gépi kezelésére az ad lehetőséget, hogy a mátrixok nagyon sok 0 mátrixelemet tartalmaznak, ritka *(sparse)* mátrixok.

A 60-as évek elejétől kezdtek megjelenni azok a dolgozatok, amelyek a ritka mátrixokkal foglalkoznak, pontosabban olyan módszerekkel amelyek ritka mátrixú lineáris egyenletrendszereket oldanak meg, invertálnak ritka mátrixokat, ill. azok sajátértékproblémáját vizsgálják. Tudományos konferenciákat rendeztek a ritka mátrixokról (lásd pl. [3], [4]).

A 60-as és a 70-es években főleg műszaki folyóiratokban a dolgozatok tömege jelent meg, amelyekben ritkamátrixos módszereket dolgoznak ki a legkülönbözőbb műszaki feladatok megoldására, elsősorban hálózatok (elektromos, víz, gáz) felada-

tainak számítására. Tankönyvek jelentek meg a ritka mátrixok vizsgálatával kapcsolatos elvi kérdések tisztázására (lásd pl. [5]).

A [6] dolgozatban J. S. DUFF összefoglalja a ritka mátrixkutatás 1977-es állapotának megfelelő problémáit, eredményeit. A dolgozat nagyon jó áttekintést ad a témakörben fellépő problémákról és azokról a módszerekről amelyeket a problémák megoldására javasoltak. Vázlatosan ismertet is nagyon sok módszert. A dolgozat irodalomjegyzéke 604 irodalmi hivatkozást tartalmaz, ami önmagában is igazolja a ritkamátrixos kutatás intenzitását. Dolgozatunk tárgya olyan módszerek, programok ismertetése, amelyek ritkamátrixú lineáris egyenletrendszer megoldására, mátrixok invertálására, determinánsának kiszámítására és sajátértékek számítására használhatók.

Nagyméretű feladatoknál még a ritkaság kihasználása esetén is kicsinek bizonyulhat az operatív memória nem túl nagy számítógép esetén. Programjaink (az első kivételével) „kis gépekre" készültek. Vagyis az operatív memória mellett háttér memóriát *(diszket)* is használnak a mátrixok tárolására, de olyan szervezésűek, hogy a memóriák közti adatátviteli idők minimálisak legyenek.

A nyugati irodalomban több program ismertetése található meg ritka mátrixok kezelésére (lásd pl. [7]). Nagy számítógépek programkönyvtáraiban találhatók is ilyen programok (pl. IBM). Ezek a programok csak „nagy gépek" esetén használhatók, ugyanis többnyire csak a gép operatív memóriáját használják. Kisebb gépeken használva azokat a programokat, csak kisebb feladatok megoldására alkalmasak. Célunk olyan programcsomag összeállítása, ami kisebb gépeken is használható. A programok FORTRAN nyelven készültek a CDC 3300-as gépre. Minden programnál megadjuk a program használatához szükséges memóriaigényt is.

A lineáris egyenletrendszerek megoldásához az első négy programban eliminációt használunk. Eliminálás közben a mátrixban felléphetnek új nem zéró elemek olyan helyen is, ahol kezdetben zéró volt a mátrixelem *(„fil in")*. Ennek a figyelése, valamint az új nem zéró elem kezelése nagyon elbonyolítja a programokat. Ez az oka annak, hogy például a lineáris egyenletrendszerek megoldására ismert egyszerű programok helyett nagyon bonyolult, terjedelmes programok adódnak. Ez különösen látszik az első programon. A második programban nem használjuk ki teljesen a ritkaság adta lehetőségeket, és a program dolgozik bizonyos helyeken zérusokkal is. Ezáltal viszont egyszerűsödik a program.

2. A feladatok és megoldási módszereik

Ebben a pontban ismertetjük a feladatokat, a megoldásukra alkalmazott numerikus módszereket. A feladatokban adott $\mathbf{A}=\{a_{ij}\}$, $i,j=1,\ldots,n$ ritka mátrix és $\mathbf{b}=\{b_i\}$ vektor esetén megoldandó az $\mathbf{Ax}=\mathbf{b}$ lineáris egyenletrendszer, vagy invertálandó az $\mathbf{A}$ mátrix, vagy számolandók annak sajátértékei. Az első programban számoljuk még a det($\mathbf{A}$) determinánst is.

1. Feladat. Megoldandó az $\mathbf{Ax}=\mathbf{b}$ egyenletrendszer, vagy számolandó az $\mathbf{A}^{-1}$ inverz. A megoldásra a *Crout-féle módszert* használjuk. Ez a következő:

Buntsuk fel az $\mathbf{A}$ mátrixot $\mathbf{A}=\mathbf{LR}$ szorzatra, ahol $\mathbf{L}$ alsó, $\mathbf{R}$ felső háromszögmátrix. Ekkor az egyenletrendszer

$$\mathbf{Ax} = \mathbf{LRx} = \mathbf{b}, \tag{2.1}$$

amit az $\mathbf{y}=\mathbf{Rx}$ változó bevezetésével két egyenletrendszer egymás utáni megoldására vezethetünk vissza:

$$\mathbf{Ly} = \mathbf{b}, \quad \mathbf{Rx} = \mathbf{y}. \tag{2.2}$$

Minthogy $\mathbf{L}$ és $\mathbf{R}$ is háromszögmátrixok, ezért a (2.2) megoldása visszahelyettesítéssel végezhető el, ami egyszerű és gyors.

Az $\mathbf{A}$ mátrix invertálásához az $\mathbf{AX}=\mathbf{I}$ egyenletrendszert oldjuk meg *Crout-féle módszerrel,* ahol $\mathbf{I}$ az egységmátrix. A szorzatra bontás után megoldandó az

$$\mathbf{AX} = \mathbf{LRX} = \mathbf{I} \tag{2.3}$$

egyenletrendszer a következő egyenletrendszerek egymásutáni megoldásával:

$$\mathbf{LY} = \mathbf{I}, \quad \mathbf{RX} = \mathbf{Y}.$$

Vagyis az invertálást úgy végezzük el, hogy először elvégezzük az $\mathbf{A}=\mathbf{LR}$ felbontást, majd rendre az egységmátrix oszlopait egy lineáris egyenletrendszer jobb oldalának tekintve elvégezzük a (2.2) visszahelyettesítést. Ily módon az inverz mátrixot oszloponként kapjuk meg.

Megjegyezzük, hogy még a nagyon ritka mátrix inverze is lehet teljesen „tele" mátrix. Szemléltető példa erre az a háromátlós mátrix, amelynek a főátlóban 2-es, a főátló alatt és felett -1-es áll. Ezen mátrix inverzének minden elme 0-tól különböző. Ezért a ritka mátrixok invertálásánál nem törekedhetünk arra, hogy a teljes inverz mátrixot tároljuk a gyorsmemóriában. A fent javasolt megoldás elkerüli ezt a problémát azáltal, hogy az inverz mátrixnak egyszerre csak egy oszlopát számítja ki.

A program a mátrix $\mathbf{A}=\mathbf{LR}$ felbontását *Gauss eliminációval* végzi. Az elimináció közben vizsgálja a szingularitást és szükség esetén sorcserét hajt végre. Közben számítja a mátrix determinánsát is.

2. Feladat. Az $\mathbf{Ax}=\mathbf{b}$ lineáris egyenletrendszer megoldása. A feladat megoldására diszk memória segítségével a *Gauss eliminációt* használjuk.

A *Gauss elimináció* használata közben a mátrix főátlója kitüntetett szerepet játszik. Ritka mátrixok esetén az új nem záró elemek várható fellépésének helyeit is a főátlóhoz igazodva előre az elimináció megkezdése előtt ki tudjuk jelölni (lásd [8]).

Ezt a lehetőséget felhasználjuk a programban is. Ezáltal ugyan dolgozik a program bizonyos zéró elemekkel is, de a program egyszerűbbé válik.

Tekintsük az i-edik sor főátlótól balra eső részét. Legyen az i-edik sor első nem zéró eleme a j-edik oszlopban. Az elimináció folytán az i-edik sor főátlótól balra eső részében új nem zéró elem csak j-nél nagyobb indexű oszlopában léphet fel. Hasonlóan legyen a j-edik oszlop főátló feletti részében az első nem zéró elem az i-edik sorban. Az elimináció közben az i-edik oszlop főátló feletti részében új nem zéró elem csak az i-nél nagyobb indexű sorban léphet fel. Ily módon a mátrixban az elimináció megkezdése előtt kijelölhetjük azokat a helyeket, ahol nem zéró elemek vannak, vagy új nem zéró elemek léphetnek fel. A második programunkban ezt a meggondolást alkalmazzuk, és a mátrixnak az ily módon előre „kijelölt" részét használjuk a számítás közben.

Feltételezzük, hogy a mátrixunk nagy, és még a fentiekben kijelölt része sem fér el a gépünk operatív memóriájában. A mátrixot sorok mentén részekre bontjuk, úgy, hogy minden részből kettő elférjen egyszerre az operatív memóriában. A részmátrixo-

kat diszken tároljuk, részenként visszük át az operatív memóriába és ott a részek közt végzünk eliminációt a következő sorrendben. Eliminálunk az első részében, majd az eliminált első rész segítségével eliminálunk a második részben. Ezután az eliminált első és második rész segítségével eliminálunk a harmadik részben, stb. Minden eliminált részt visszaírjuk a diszkre. Végül ezek segítségével visszahelyettesítést végzünk és így kapjuk a megoldást. Ennek a módszernek mátrixinvertálásra való alkalmazásával foglalkozik a (2) dolgozat.

3. Feladat. Megoldandó az $\mathbf{Ax}=\mathbf{b}$ lineáris egyenletrendszer szalagmátrix esetben, vagyis legyen

$$a_{ij} = 0, \quad \text{ha} \quad |i-j| > k.$$

A megoldást diszk segítségével végezzük *Gauss eliminációval.*

1. Megoldás.

Az elimináció megkezdése előtt a szalagmátrix első

$$\mathbf{A}_1 = \{a_{ij}\}, \quad i, j = 1, \ldots, k+1$$

részét tároljuk az operatív memóriában, a szalagmátrix többi részét diszken tároljuk a következő sorrendben: az első rekordban az a_{ij}, $(i=k+1, j=2, \ldots, k+1)$, $(j=k+1, i=k, k-1, \ldots, 2)$ mátrixelemeket; a második rekordban az a_{ij}, $(i=k+2, j=3, \ldots, k+2)$, $(j=k+2, i=k+1, k, \ldots, 3)$ mátrixelemeket stb.

Elvégzünk egy eliminációs lépést, vagyis az $\mathbf{A}_1$ első sora segítségével eliminálju az $\mathbf{A}_1$ első oszlopát. Ezután az $\mathbf{A}_1$ első sorát diszkre visszük, az első oszlopát elhagyjuk és egyesítjük $\mathbf{A}_1$-hez az első rekordot a diszkről. Az új $(k+1)\times(k+1)$-es részmátrix kapja meg az $\mathbf{A}_1$ szerepet, és végezzük el rajta az előbb leírt lépéseket. Ezt addig ismételjük, míg az egész szalagmátrix eliminálásra nem kerül. Ezután a minden egyes elimináció után diszkre vitt részek segítségével visszahelyettesítést végzünk, és ezzel megkapjuk a megoldást.

2. Megoldás

A 3. feladat megoldását elvégezzük más módszerrel is. A 2. megoldás abban áll, hogy minden lépésben oszloponként főelemvizsgálatot végzünk, és ha a főelem kicsinek bizonyul sorcserét hajtunk végre. Ebben az esetben az operatív memóriában a szalagmátrix k sorát és $2k+1$ oszlopát tároljuk. Ehhez minden elimináció megkezdése előtt egy sort kapcsolunk hozzá (ez lesz a $k+1$-edik sor), illetve minden elimináció után egy sort (az első sort) diszkre írjuk. Az elimináció befejezése után a diszkre írt eliminált sorokkal visszahelyettesítést végzünk.

A főelem választást egy szabadon választott paraméterrel vezéreljük. A paraméter megválasztásával elérhetjük, hogy minden eliminációs lépés megkezdése előtt kiválaszthatjuk a főelem oszlopból a legnagyobb abszolútértékű elemet, vagy csak közel szingurális esetben csináljuk ezt. A 2. megoldással elkerülhetjük a szingularitást vagy közel szingularitási lehetőségeket, és ezáltal javíthatjuk a megoldás stabilitását.

A 3. feladat megoldására javasolt 1. megoldás gyorsabb és nagyobb méretekben használható, de hibás eredményt ad, ha az egyenletrendszer közel szinguláris. A 2. megoldás valamivel lassabban dolgozik, kisebb méretekben használható, de pontosabb megoldást szolgáltat.

4. Feladat. Megoldandó az $\mathbf{Ax}=\mathbf{b}$ lineáris egyenletrendszer szimmetrikus szalagmátrix esetén, vagyis legyen

$$a_{ij}=0, \quad \text{ha} \quad |i-j|>k \quad \text{és} \quad a_{ij}=a_{ji}.$$

A megoldást diszk segítségével végezzük *Cholesky módszer* segítségével. Az **A** mátrixot először $\mathbf{A}=\mathbf{LR}$ szorzatra bontjuk (ahol **L** és **R** háromszögmátrixok) *Cholesky módszerrel,* majd rendre visszahelyettesítésekkel megoldjuk az

$$\mathbf{Ly}=\mathbf{b}, \quad \mathbf{Rx}=\mathbf{y} \tag{2.4}$$

egyenletrendszereket. A program az operatív memóriában $k(k+1)/2$ helyet foglal el a szalagmátrix egy $\mathbf{A}_1$-es részére, a szalagmátrix többi részét diszken tárolja. A szorzatra bontás az $\mathbf{A}_1$-ben n lépésben hajtódik végre. Minden lépés előtt egy k elemet tartalmazó szalagsort egyesítünk hozzá, illetve minden eliminációs lépés után egy k elemet tartalmazó részt viszünk a diszkre. A diszkre vitt részekből épülnek fel a faktormátrixok, aminek a segítségével végezzük el a visszahelyettesítést.

5. Feladat. Sajátértékek kiszámítása szimmetrikus esetben. Alkalmazzuk először az **A** szimmetrikus ritka mátrixra a *Lánczos módszer* szimmetrikus változatát, ami a következőképpen fogalmazható meg (lásd pl. [9]): A $\mathbf{v}_1$ normált vektorból kiindulva $k=1, 2, \ldots, n$-re elvégezzük a következő számításokat:

$$\begin{aligned}
&\alpha_k = \mathbf{v}_k^T \mathbf{A}\mathbf{v}_k,\\
&\mathbf{w}_{k+1} = \mathbf{A}\mathbf{w}_k - \alpha_k \mathbf{v}_k - \beta_{k-1}\mathbf{v}_{k-1}, \quad (\beta_0 = 0),\\
&\beta_k = \sqrt{\sum_i (\mathbf{w}_{k+1})_i^2},\\
&\mathbf{v}_{k+1} = \mathbf{w}_{k+1}/\beta_k,
\end{aligned} \tag{2.5}$$

ahol $(\mathbf{w}_{k+1})_i$ a $\mathbf{w}_{k+1}$ vektor i-edik összetevője.
A (2.5)-ből számolt α_k, β_k számokból felépített

$$\mathbf{T}=\begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \beta_2 & \\ & \ddots & \ddots & \\ & & & \alpha_n \end{bmatrix} \tag{2.6}$$

mátrix hasonló az eredeti **A** mátrixhoz, vagyis a **T** és az **A** sajátértékei megegyeznek. A számolás második lépésében kiszámítjuk a **T** mátrix sajátértékeit az *implicit QR módszer* segítségével.

Ezt a megoldást azért célszerű ritkamátrixokra alkalmazni, mert a nem zéró elemek akár koordinátás tárolása esetén, akár külső tárolása esetén is a (2.5) számítások könnyen elvégezhetők. A (2.6)-ban felépített **T** mátrix viszont már háromátlós, így nagy méretek is tárolhatók.

A (2.5)-ben felépített $\mathbf{v}_k$ vektorok ortonormált vektorok. Nagy méretek esetén azonban az ortogonalitás elromlik. Ezért szükség van egy visszaortogonalizálásra (lásd [1]). Ha valamilyen k-ra a $\mathbf{v}_k$ zérussá vagy nagyon kicsivé válik, akkor egy új vektort kell választani, ami az eddigi ortogonalizált vektorokra merőleges és lehet folytatni a számolást.

6. Feladat. Sajátértékek számítása nem szimmetrikus esetben.

A nem szimmetrikus mátrixok sajátértékei általában komplexek. Azonban gyakran találkozunk olyan feladatokkal, amelyekben a feladat fizikai természetéből következik, hogy léteznek a feladatnak valós sajátértékei is. Ez a feladat az ilyen sajátértékek megkeresését tűzi ki célul.

Alkalmazzuk először az $\mathbf{R}$ mátrixra a *Lánczos módszert*, ami a következőképpen fogalmazható meg (lásd [1]): A $c_0=0$, $\mathbf{x}_0=\mathbf{y}_0=\mathbf{0}$, $\mathbf{x}_1$ és $\mathbf{y}_1$ tetszőleges ($\mathbf{y}_1\mathbf{x}_1\neq 0$) szám, illetve vektorokból kiindulva $k=1, \ldots, n$-re végezzük el a következő számításokat

$$
\begin{aligned}
b_k &= \mathbf{y}_k^T\mathbf{A}\mathbf{x}_k/\mathbf{y}_k^T\mathbf{x}_k,\\
\mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k - b_k\mathbf{x}_k - c_{k-1}\mathbf{x}_{k-1},\\
\mathbf{y}_{k+1} &= \mathbf{A}^T\mathbf{y}_k - b_k\mathbf{y}_k - c_{k-1}\mathbf{y}_{k-1},\\
c_k &= \mathbf{y}_{k+1}^T\mathbf{x}_{k+1}/\mathbf{y}_k^T\mathbf{x}_k.
\end{aligned}
\tag{2.7}
$$

A (2.7)-ben számolt b_k, c_k számokból építsük fel a

$$
\mathbf{T}=\begin{vmatrix} b_1 & c_1 & & \\ 1 & b_2 & \ddots & \\ & \ddots & \ddots & c_2 \\ & & & b_n \end{vmatrix}
\tag{2.8}
$$

mátrixot, ami hasonló az $\mathbf{A}$ mátrixhoz, vagyis az $\mathbf{A}$ és a $\mathbf{T}$ sajátértékei megegyeznek. A számolás második részében kiszámítjuk a $\mathbf{T}$ mátrix sajátértékeit *„Bisetion" módszerrel* (lásd [10]). A [10]-ben bizonyított tétel szerint a $\mathbf{B}$ mátrix p-nél kisebb sajátértékeinek száma megegyezik az

$$
1,\quad \det(\mathbf{B}_1-p\mathbf{I}_1),\ \det(\mathbf{B}_2-p\mathbf{I}_2),\ \ldots,\ \det(\mathbf{B}_n-p\mathbf{I}_n)
\tag{2.9}
$$

sorozatban az előjelváltozások számával. (2.9)-ben $\mathbf{B}_i$ és $\mathbf{I}_i$ jelenti a $\mathbf{B}$, illetve az $\mathbf{I}$ egységmátrix bal felső $i\times i$-s minormátrixát. A (2.9) sorozat számítása a (2.8) mátrixra alkalmazva nagyon gyors.

A (2.7)-ben képződő $\mathbf{x}_k$ és y_k vektorok biortogonálisok. Azonban nagy méretek esetén a biortogonalitás elromolhat. Ebben az esetben új $\mathbf{x}_k$ és $\mathbf{y}_k$ vektorokat kell választani úgy, hogy $\mathbf{x}_k$ ortogonális legyen az addigi $\mathbf{y}_i$ vektorokra, illetve $\mathbf{y}_k$ ortogonális legyen az addigi $\mathbf{x}_i$ vektorokra. Ezután a számolást lehet folytatni.

3. A programok ismertetése

Most ismertetjük a 2. részben megfogalmazott 6 feladat megoldásainak programjait. A programok szerkezeti felépítései input adatai és outputjai is különböznek, ezért ezeket külön-külön ismertetjük majd.

A programokat FORTRAN nyelven a CDC 3300-as gépre írtuk. A programok használatához szükséges helyszükségletet a CDC gépnél használatos CORE-ban és segmens-ben adtuk meg:

1 qp *(quarter page)* = 512 24 bites szó, míg 20 qp = 1 seg(segmens) a diszk-en.

1. Program. Ritkamátrixú lineáris egyenletrendszer megoldása vagy mátrix invertálás (lásd az 1-es feladat). A főprogram beolvassa kártyáról az N, IP, NY, EPS paramétereket (3I5, E15.7) formátummal, majd a nem zéró a_{ij} mátrixelemeket és azok sor és oszlop indexeit 3(2I5, E16.7) formátummal a 10-es file-ról, (negatív sorindex a 10-es file-on jelzi az adatok végét), majd kiírja az input adatokat. Felépíti a diagonál elemekből az A0 tömböt, a diagonálon kívüli elemekből az *A*1 tömböt, azok sor, illetve oszlop indexeiből pedig az M1 tömböt. A program dinamikus tárolási módot alkalmaz, ami *„linked list"* elven működik, (lásd [5]). Az M1 egész értékű tömbben minden nem zéró elemhez két egész szám tartozik. Az első a nem zéró elem oszlop (illetve sor) indexe, a második egy pointer, ami mutatja az ugyanebben a sorban (vagy oszlopban) következő nem zéró elem helyét az A1 tömbben. Ha egy sorban, (illetve oszlopban) nincs több nem zéró elem, akkor a pointer érték zéró marad. A sorokban (illetve oszlopokban) fellépő első nem zéró elem helye az A1 tömb 1-től N-ig (illetve N+1-től 2N−1 ig) terjedő helye. Az A1 és az M1 tömbök COMMONba kerülnek. Ezután a program hívja a paraméterektől függően a szubrutinokat, majd kinyomtatja az NY paramétertől függően a kívánt eredményeket.

Paraméterek jelentése:

N:	a mátrix rendszáma;
IP=0:	**A**=**LR** faktorizálás;
IP=−1:	**A**=**LR** faktorizálás és az $\mathbf{A}^{-1}$ számolása és oszloponkénti kinyomatatása;
IP=m:	**A**=**LR** faktorizálás, majd *m*-szer a lineáris egyenletrendszerek jobb oldalainak beolvasása, az egyenletrendszerek megoldása és az eredmények kinyomtatása;
IP<−1:	kinyomtatódik a mátrix:
NY=0:	a mátrix kinyomtatása a faktorizálás előtt és után is;
NY=1:	a mátrix kinyomtatása a faktorizálás előtt;
NY=2;	a mátrix kinyomtatása a faktorizálás után;
NY=3:	a mátrix nem nyomtatódik ki;
EPS:	a szingularitást vizsgálja. Ha *k*-adik lépésben a főátló abszolút értékben kisebb mint EPS, akkor a főátló oszlopából lenagyobb elemet keresi és sorcserét hajt végre. Ha a mátrix szinguláris, erről jelzést ad és leállítja a számolást.

A programban használt szubrutinok:

LR végzi a faktorizálást;
CS végzi a főelemválasztást és a sorcserét;
NYTAB nyomtatja ki a mátrixot. A mátrix nyomtatása az A0, A1 és M1 tömbök kinyomtatását jelenti;
VISZ végzi a visszahelyettesítéseket;
NYX az eredmények kinyomtatását végzi.

A CDC gépen kipróbált program 84 qp felhasználásával a következő méretű feladat megoldását teszi lehetővé; $N \leqq 1000$, a nem zéró elemek száma $\leqq 5500$, az egy sorban levő nem zéró elemek száma $\leqq 200$.

A program listája:

```
C
C  THE PROGRAM SOLVES THE SYSTEM OF LINEAR EQUATIONS A*X=B WITH SPARSE
C     MATRIX, OR INVERTS A SPARSE MATRIX BY COLUMNS. THE PROGRAM COMPUTS
C     THE DETERMINANT OF THE MATRIX IN BOTH CASES.
C  METHOD — CROUT'S. THE DIAGONAL ELEMENTS ARE IN AFFAY AO, THE NON
C     DIAGONAL ELEMENTS ARE IN ARRAY A1, THEIR RO W OR COLUMN INDEXES
C     ARE IN ARRAY M1.
C
C  PARAMETERS:

C     N — NUMBER OF EQUATIONS;
C     IP=0 — THE MATRIX IS FACTORIZED TO THE PRODUCT FORM A=L*R;
C     IP=−1 — THE MATRIX IS FACTORIZED TO THE PRODUCT FORM, THE INVERSE
C              OF THE MATRIX A IS COMPUTED AND WRITTEN OUT BY COLUMNS.
C     IP=M — THE MATRIX IS FACTORIZED TO THE PRODUCT FORM A=L*R.
C              M TIMES: THE RIGHT—HAND—SIDE IS READ, THE EQUATION IS SOLVED,
C              THE SOLUTIONS ARE WRITTEN OUT;
C     IP. LT. −1 — ONLY THE MATRIX IS WRITTEN OUT.
C     THE PARAMETER NY CONTROLS THE OUTPUT. THE MATRIX IS WRITTEN OUT IF:
C     NY=0 BEFORE AND AFTER THE ELIMINATION;
C     NY=1 BEFORE THE ELIMINATION;
C     NY=2 AFTER THE ELIMINATION;
C     NY=3 NO WRITTING OUT.

C  EPS CONTROLS THE SINGULARITY

C
      DIMENSION I1(5), J1(5),
     1 A0(1000), M21(1000), M11(1000), B(1000), X(1000), MS(1000)
      COMMON /20/ A1(5500), M1(11000)
      REWIND 1
      READ 1, N, I, P, NY, EPS
    1 FORMAT (3I5/, E15.7)
      PRINT 1, N, IP, NY, EPS
      DO 9 I=1, N
      MS(I)=I
      M11(I)=I
      IN=I+N
    9 M21(I)=IN
      M3=2*N
    8 READ(1,10) ((I1(I), J1(I), B(I)), I=1,3)
      PRINT 2, ((I1(I), J1(I), B(I)), I=1,3)
    2 FORMAT (1H, 3(2I5, E16.7))
```

```
 10 FORMAT (3(2I5, E16.7))
      DO 4 K=1,3
      IF(I1(K)) 99,4,3
    3 I=I1(K)
      J=J1(K)
      IF(I—J) 5,6,7
    6 A0(I)=B(K)
      GOTO 4
    5 M=M11(I)
      A1(M)=B(K)
      M11 (I)=M3
      M1 (2*M)=J
      M1 (2*M—1)=M3
      M3=M3+1
      GOTO 4
    7 M=M21 (J)
      A1 (M)=B(K)
      M21(J)=M3
      M1 (2*M)=I
      M1 (2*M—1)=M3
      M3=M3+1
    4 CONTINUE
      GOTO 8
   99 CONTINUE
      IF (IP.GT.—2) GOTO 11
      CALL NYTAB (N,M3,A0)
      GOTO 98
   11 CONTINUE
      IF(NY.GT.1) GOTO 90
      CALL NYTAB(N, M3,A0)
   90 CONTINUE
      CALL LR(N,M3,A0,M11,M21,EPS,MS,B,X)
      IF(NY.EQ.1.OR.NY.EQ.3) GOTO 91
      CALL NYTAB(N,M3,A0)
   91 CONTINUE
      IF(IP.EQ.0) GOTO 98
      IF(IP.GT.0) GOTO 20
      DO 39 M=1,N
      DO 31 I=1,N
   31 B(I)=0.
      MM=MS(M)
      B(MM)=1.
      CALL VISZ(N,MM,IP,A0,B,X)
      CALL NYX(N,M,X)
   39 CONTINUE
   20 REWIND 2
      DO 29 M=1, IP

C
C  THE RIGHT HAND SIDE IS READING
C
      READ(2,21) (B(I),I=1,N)
   21 FORMAT (5E15.5)
      CALL VISZ(N,M,IP,A0,B,X)
      CALL NYX(N,M,X)
   29 CONTINUE
   98 CONTINUE
      END
      SUBROUTINE LR(N,M3,A0,M11,M21,EPS,MS,A5,A6)
```

```
C
C  THE SUBROUTINE LR FACTORIZED THE MATRIX TO THE PRODUCT FORM A=L*R
C
      DIMENSION A0(1),M11(1),M21(1),MS(1),A5(1),A6(1),
     1 M6(200),M5(200)
      COMMON /20/ A1 (5500),M1(11000)
      N1=N-1
      DET=1.
      DO 10 K=1,N1
C
C  IF THE PIVOT ELEMENT IN MODUL .LE. EPS THEN NEW PIVOT ELEMENT IS
C  CHOSEN BY COLUMN AND EXCHANGE OF ROWS IS DONE.
C
      IF (ABS(A0(K)).GT.EPS) GO TO 19
      CALL  CS(K,N,M3,A0,M11,M21,MS,EPS)
      CALL CS(K,N,M3,A0,M11,M21,MS,EPS)
      DET=-DET
   19 K1=M1(2*K-1)
      K5=K
      K2=1
    1 IF(K1.EQ.0) GOTO 2
      IF(M1(2*K5).LE.K) GOTO 22
      M5(K2)=M1(2*K5)
      A1(K5)=A1(K5)/A0 (K)
      A5(K2)=A1(K5)
      K2=K2+1
   22 K5=K1
      K1=M1(2*K1-1)
      GOTO 1
    2 KA=K+N
      K1=M1(2*KA-1)
      K5=KA
      K3=K2-1
      K2=1
    6 IF(K1.EQ.0) GOTO 5
      M6(K2)=M1(2*K5)
      A6(K2)=A1(K5)
      K5=K1
      K1=M1(2*K1-1)
      K2=K2+1
      GOTO 6
    5 K4=K2-1
      DO 7 I=1,K3
      DO 7 J=1,K4
      MI=M6(J)
      MJ=M5(I)
      IF(MI.NE.MJ) GOTO 9
      A0(MI)=A0(MI)-A5(I)*A6(J)
      GOTO 7
    9 IF(MI.GT.MJ) GOTO 11
      K5=MI
      K1=M1(2*MI)
      K2=M1(2*MI-1)
   13 IF(K2.EQ.0) GOTO 14
      IF(K1.EQ.MJ) GOTO 12
      K5=K2
      K1=M1(2*K2)
      K2=M1(2*K2-1)
      GOTO 13
```

```
   12 A1(K5)=A1(K5)-A5(I)*A6(J)
      GOTO 7
   14 L=M11(MI)
      A1(L)=-A5(I)*A6(J)
      M1(2*L)=MJ
      M1(2*L-1)=M3
      M11(MI)=M3
      M3=M3+1
      GOTO 7
   11 MA=MJ+N
      K1=M1(2*MA)
      K2=M1(2*MA-1)
      K5=MA
   15 IF(K2.EQ.0) GOTO 16
      IF(K1.EQ.MI) GOTO 17
      K5=K2
      K1=M1(2*K2)
      K2=M1(2*K2-1)
      GOTO 15
   17 A1(K5)=A1(K5)-A5(I)*A6(J)
      GOTO 7
   16 L=M21(MJ)
      A1(L)=-A5(I)*A6(J)
      M1(2*L)=MI
      M1(2*L-1)=M3
      M21(MJ)=M3
      M3=M3+1
    7 CONTINUE
   10 DET=DET*1A0(K)
      DET=DET*A0(N)
      PRINT 18,DET
   18 FORMAT(////7H DET=,E15.7)
      RETURN
      END
      SUBROUTINE CS(K,N,M3,A0,M11,M21,MS,EPS)
C
C  THE SUBROUTINE CS PERFORMS THE EXCHANGE OF RCWS.
C
      DIMENSION A0(1),M11(1),M21(1),MS(1)
      COMMON /20/ A1(5500),M1(11000)
      KK=K
      A=ABS(A0(K))
      J=K+N
    2 J1=M1(2*J-1)
      IF(J1.EQ.0) GOTO 3
      IF(A.GE.ABS(A1(J))) GOTO 1
      A=ABS(A1(J))
      KK=1
    1 J=J1
      GOTO 2
    3 IF (A.LT.EPS) GOTO 21
C
C  THE PIVOT ELEMENT IS CHOSEN FROM J2-TH ROW.
C
      J2=M1(2*KK)
      AK=A0(K)
      A0(K)=A1(KK)
      A1(KK)=AK
      M1(2*KK)=J2
```

```
      MS1=MS(K)
      MS(K)=MS(J2)
      MS(J2)=MS1
      AJ=A0(J2)
      A0(J2)=0
      K1=K-1
      DO 4 I=1, K1
      I4=I+N
    6 I1=M1(2*I4-1)
      I3=0
      IF(I1.EQ.0) GOTO 4
      IF(M1(2*I4).EQ.K) I3=I4
      IF(M1(2*I4).EQ.J2) M1(2*I4)=K
      IF(I3.NE.0) M1(2*I4)=J2
      I4=I1
      GOTO 6
    4 CONTINUE
      I=K
   11 I1=M1(2*I-1)
       IF(I1.EQ.0) GOTO 10
      I2=M1(2*I)
      IF(I2-J2) 7,8,13
    8 A0(J2)=A1(I)
   13 I=I1
      GOTO 11
    7 I4=I2+N
   15 I5=M1(2*I4-1)
      IF(I5.EQ.0) GOTO 19
      IF(M1(2*I4).NE.J2) GOTO 14
      A=A1(I4)
      A1(I4)=A1(I)
      I5=M11(J2)
      M11(J2)=M3
      M1(2*I5-1)=M3
      M3=M3+1
      M1(2*I5)=I2
      A1(I5)=A
      GOTO 13
   19 I5=M21(I2)
      M21(I2)=M3
      M1(2*I5-1)=M3
      M3=M3+1
      M1(2*I5)=J2
      A1(I5)=A1(I)
      GOTO 13
   14 I4=M1(2*I4-1)
      GOTO 15
   10 I5=M11(J2)
      M11(J2)=M3
      M1(2*I5-1)=M3
      M1(2*I5)=J2
      M3=M3+1
      A1(I5)=AJ
      K1=K+1
      K2=J2-1
      DO 23 K3=K1,K2
      J=K3+N
      J1=K
```

```
   25 IF(M1(2*J-1).EQ.0) GOTO 23
      IF(M1(2*J).NE.J2) GOTO 24
   27 IF(M1(2*J1).EQ.K3) GOTO 23
      IF(M1(2 J1).GE.J2) GOTO 31
      IF(M1(2*J1-1).EQ.0) GOTO 26
   31 J1=M1(2*J1-1)
      GOTO 27
   24 J=M1(2*J-1)
      GOTO 25
   26 I5=M11(J2)
      M11(J2)=M3
      M1(2*I5-1)=M3
      M3=M3+1
      M1(2*I5)=K3
      A1(I5)=A1(J)
      A1(J)=0
   23 CONTINUE
      L1=M1(2*K-1)
      L2=M1(2*K)
      M1(2*K-1)=M1(2*J2-1)
      M1(2*K)=M1(2*J2)
      M1(2*J2-1)=L1
      M1(2*J2)=L2
      A=A1(K)
      A1(K)=A1(J2)
      A1(J2)=A
      I=J2
   28 I1=M1(2*I)
      IF(I1.EQ.0) GOTO 29
      IF(I1.GT.J2) GOTO 32
      A1(I)=0
   32 I=M1(2*I-1)
      GOTO 28
   29 CONTINUE
      GOTO 16
   21 CONTINUE
      PRINT 20,N,K
   20 FORMAT (12H SINGULARIS,2I5//)
      STOP
   16 CONTINUE
C
C  THE MARK OF EXCHANGE.
C
      PRINT 33,K,J2
   33 FORMAT(7H CSERE:,2I5)
      I=M11(J2)
      M11(J2)=M11(K)
      M11(K)=I
      RETURN
      END
      SUBROUTINE NYTAB(N,M3,A0)
C
C  THE SUBROUTINE NYTAB EXECUTES THE OUTPUT OF MATRIX
C
      DIMENSION A0(1)
      COMMON /20/ A1(5500),M1(11000)
```

```
      M5=2*M3
      PRINT 1,N
      PRINT 2,(A0(I),I=1,N)
      PRINT 2,(A1(I),I=1,M3)
      PRINT 3,(I,I)=1,10)
      PRINT 1,(M1(I),I=1,M5)
    3 FORMAT(/1019/)
    1 FORMAT(10(I5,I4))
    2 FORMAT(1H,10F10.5)
      RETURN
      END
      SUBROUTINE VISZ(N,M,IP,A0,B,X)
C
C  THE SUBROUTINE VISZ EXECUTES THE BACK-SUBSTITUTION
C
      DIMENSION A0(1),B(1),X(1)
      COMMON /20/ A1(5500),M1(11000)
      DO 1 I=1,N
    1 X(I)=0
      K=1
      IF(IP.LT.0) K=M
      DO 2 J=K,N
      X(J)=B(J)/A0(J)
      J3=J+N
    3 J1=M1(2*J3-1)
      J2=M1(2*J3)
      IF(J1.EQ.0) GOTO 2
      B(J2)=B(J2)-A1(J3)*X(J)
      J3=J1
      GOTO 3
    2 CONTINUE
      N1=N-1
      DO 5 I=1,N
    5 B(I)=X(I)
      DO 4 I=1,N1
      I1=N-I
      J=I1
    6 J1=M1(2*J-1)
      J2=M1(2*J)
      IF(J1.EQ.0) GOTO 4
      B(I1)=B(I1)-A1(J)*X(J2)
      J=J1
      GOTO 6
    4 X(I1)=B(I1)
      RETURN
      END
      SUBROUTINE NYX(N,M,X)
C
C  THE SUBROUTINE NYX PRINTS THE SOLUTION OF EQATION A*X=B.
C
      DIMENSION X(1)
      PRINT 1,M
      PRINT 2,(X(I),I=1,N)
    1 FORMAT(//I5/)
    2 FORMAT(5E18.9)
      RETURN
      END
```

2. Program. Ritka mátrixú lineáris egyenletrendszer megoldása diszk segítségével (lásd a 2-es feladat).

A főprogram beolvasása a 10-es file-ről az N, N9 és EPS paramétereket (2I9, E15.5) formátummal.

Ezután olvassa a 20-as file-ról 3(E16.9, 2I5) formátummal a ritka mátrixnak nem zérus elemeit, azok sor és oszlop indexeit. A nem zéró elemeket soronként egymásután rendezve kell a 20-as file-ra elhelyezni. Ha ettől eltérő nem zéró elem is előfordul az nem kerül feldolgozásra. Erről hibajelzést ad a program. Negatív sorindex jelöli a mátrixelemek tömbjének a végét. Végül olvassa a 30-as file-ról az egyenletrendszer jobboldalát (5E15.9)-es formátummal.

A főprogram az adatok beolvasása közben a RENDEZ szubrutin segítségével felépíti az 1-es diszk file-on azokat a részmátrixokat, amelyek a számoláshoz kellenek. Az eliminációt és a visszahelyettesítést a SLED szubrutin végzi. Az egyenletrendszer megoldása után az eredmények kinyomtatódnak.

Paraméterek jelentése:

N: az egyenletek (és az ismeretlenek) száma;
N9: egy részmátrix maximális mérete;
EPS: a szingularitást vizsgáló paraméter.

A CDC gépen 84 qp és 50 seg diszkterület felhasználásával a következő méretű feladat oldható meg:
$N \leqq 1000$. $N9 \leqq 4000$, a mátrix felbontásának száma $\leqq 100$.

A program listája:

```
C
C  THE PROGRAM SOLVES THE SPARSE LINEAR SYSTEM WITH HELP OF THE
C      SUBROUTINE SLED. THE PROGRAM CARRIES OUT THE INPUT FROM
C      FILE 20, BY MEANS OF SUBROUTINE RENDEZ
C      CONSTRUCTS THE MATRIX IN DISC, CALLS THE SOUBROUTINE SLED AND
C      PRINTS THE SOLUTION.

C      PARAMETERS:

C      N — THE NUMBER OF EQUATIONS;
C      N9 — MAX LENGTH OF A PART OF THE MATRIX;
C      EPS — THE PARAMETER FOR INVESTIGATION OF THE SINGULARITY

C  THE PROGRAM USES THE FOLLOWING ARRAYS;

C      M1(I) — THE NUMBER OF NONZERO ELEMENTS LEFT TO THE DIAGONAL IN THE
C              I—TH ROW;
C      M2(I) — THE NUMBER OF NONZERO ELEMENTS RIGHT TO THE DIAGONAL IN
C              THE I—TH ROW;
C      ID(K)+1 IS THE FIRST ROW INDEX IN THE K—TH PART OF THE MATRIX;
C      ID(K+1) IS THE LAST ROW NUMBER IN THE K — TH PART OF THE MATRIX;
C      M(J): THE COLUMN INDEX OF THE NONZERO ELEMENTS RIGHT TO THE
C              DIAGONAL, SUCCESSIVELY BY ROWS.
C
C  THE PROGRAM USES SUCH A PART OF MATRIX IN WHICH THE ELEMENTS ARE NON-
C      ZERO, OR NONZERO ELEMENTS CAN APPEAR DÜRING THE ELIMINATION.
C
      DIMENSION M1(1000),M2(1000),M3(1000),M4(1000),M(3000),ID(101),I3(
      13),J1(3),IM(101),IA(101)
      2,A(4000),B(1000),X(1000)
```

```
      COMMON /10/ A,B,X,M
      REWIND 10
      READ(10,11) N,N9,EPS
      PRINT 11,N,N9,EPS
      REWIND 20
      CALL RENDEZ (N,N9,M1,M2,M3,M4,ID,I3,J1,IM,IA)
      REWIND 30
      IM1=1
      M3(1)=1
      M3(1)=1
      M4(1)=1
      DO 41 I=2,N
      M3(I)=M3(I-1)+M2(I-1)+M1(I)+1
   41 CONTINUE
      M41=2
      DO 112 I=2,N
      IF (I.LE.ID(M41)) GOTO 114
      M4(I)=1
      M41=M41+1
      GOTO 112
  114 M4(I)=M4(I-1)+M2(I-1)
  112 CONTINUE
      DO 34 K=1.N9
   34 A(K)=0
C
C     INPUT OF THE RIGHT SIDE. PERFORMATION BY FORMAT 13.
C
      READ(30,13) (B(I),I=1,N)
      ENDFILE 1
      REWIND 1
      ENDFILE 3
      REWIND 3
      ENDFILE 4
      REWIND 4
      CALL SLED(N,ID,M1,M2,M3,M4,N9,IM,IA,EPS)
C
C     OUTPUT OF THE SOLUTION
C
      PRINT 27
      PRINT 30,(I,X(I),I=1,N)
   27 FORMAT(14HITHE SOLUTION://)
   11 FORMAT(2I8,E15.5)
   13 FORMAT (5(E15.9))
   30 FORMAT(I5,E18.9)
      END
      SUBROUTINE RENDEZ (N,N9,M1,M2,M3,M4,ID,I3,J1,IM,IA)
      DIMENSION M1(1),M2(1),M3(1),M4(1),ID(1),I3(1),J1(1),IM(1),IA(1)
      COMMON /10/ A(4000),B(1000),X(1000),M(3000)
C
C  THE SUBROUTINE RENDEZ BUILTS UP THE MATRIX IN DISC AND THE ARRAYS
C     M1,M2,ID,IA AND IM BY MEANS OF INPUT.
C
      L=1
      ID1=1
      ID(ID1)=L-1
      K=0
      KA=0
      I8=0
      K3=1
```

```
C
C  INPUT OF NONZERO ELEMENTS: A(I,J), I, J, WHERE I IS THE ROW, J IS
C      THE COLUMN INDEX . PERFORMATION BY FORMAT 9. THE END MARK IS I<0.
C
    6 READ (20,9) (X(I),I3(I),J1(I),I=1,3)
      PRINT 9,(X(I),I3(I),J1(I),I=1,3)
      DO 1 I=1,3
      IF (I3(I)) 25,1,2
    2 IF (I3(I)-L) 19,3,10
    3 J2=J1(I)
      J3=I3(I)-J2
      IF(J3) 5,4,4
    4 B(J2)=X(I)
      IF (M1(L).LT.J3) M1(L)=J3
      GOTO 1
    5 K1=L+K3
      I8=I8+1
      K3=K3+1
      B(K1)=X(I)
      M3(K1)=J2
    1 CONTINUE
      GOTO 6
   10 M2(L)=I8
      K5=K+1
      L1=L+1
      L2=M1(L)+1
      DO 7 J=1,L2
      KA1=KA+J
      KB=L-M1(L)+J-1
    7 A(KA1)=B(KB)
      KA=KA+L2
      I8=0
      GOTO 26
   19 PRINT 18,L,I3(I),J1(I)
      GOTO 1
   25 I9=-1
      GOTO 10
   26 I4=L+1
      J4=L
      IF(M4(L).EQ.L) J4=L+1
   14 IF(M3(I4).EQ.0.AND.M4(J4).EQ.0) GOTO 8
      K=K+1
      IF (M3(I4) - M4(J4)) 11,12,13
   12 M(K)=M3(I4)
      KA=KA+1
      A(KA)=B(I4)
      I4=I4+1
      J4=J4+1
      GOTO 14
   13 IF (M4(J4).NE.0) GOTO 15
   16 KA=KA+1
      A(KA)=B(I4)
      M(K)=M3(I4)
      I4=I4+1
      GOTO 14
   15 M(K)=M4(J4)
      KA=KA+1
```

```
      J4=J4+1
      GOTO 14
   11 IF(M3(I4).EQ.0) GOTO 15
      GOTO 16
    8 DO 17 J=L,N
      M3(J)=0
   17 M4(J)=0
      L=L+1
      DO 20 J=K5,K
      L2=L1+J-K5
   20 M4(L2)=M(J)
      K3=1
      DO 33 J=1,N
   33 B(J)=0.
      IF(KA.LT.N9. AND I9.EQ.0) GOTO 3
      IA(ID1)=KA
      IM (ID1)=K
      WRITE (1) (A(J),J=1,KA)
      WRITE (3) (M(J),J=1,K)
      WRITE (4) (M(J),J=1,K)
      DO 24 J=1,KA
   24 A(J)=0
      ID1=ID1+1
      ID(ID1)=L-1
      DO 28 J=1,K
   28 M(J)=0
      K=0
      KA=0
      K3=1
      IF(L.LT.N.AND.09.EQ.0) GOTO 3
    9 FORMAT (3(E16.9,2I5))
   18 FORMAT (12H ORDER ERROR,3I5)
      RETURN
      END
      SUBROUTINE SLED(N,ID,M1,M2,M3,M4,N9,IM,IA,EPS)
C
C  THE SUBROUTINE SLED SOLVES THE SPARSE LINEAR SYSTEM, IT PERFORMS
C     THE GAUSS ELIMINATION AMONG PARTS OF THE SPARSE MATRIX, THE WHOLE
C     MATRIX IS IN DISC , PARTS ARE TRANSFERRED BETWEEN DISC AND MAIN
C     MEMORY.
C
      DIMENSION M1(1),M2(1),M3(1),M4(1),ID(1)
     1,A(4000),B(1000),X(1000),D(4000)
     2,MA(3000),MD(3000),IM(1),IA(1)
      COMMON /10/ A,B,X,MA
      J1=1
      IM1=1
      IM2=1
      J2=0
    1 IA=IA(IM1)
      READ(1) (A(I),I=1,IA1)
      IM3=IM(IM1)
      IM1=IM1+1
      READ (3) (MA(I),I=1,IM3)
      I3=ID(J1)+1
```

```
      I4=ID(J1+1)
      IF(J1.EQ.1) GO TO 78
      J2=M3(I3-1)+M2(I3-1)
   78 J5=J1-1
      IF (J5) 3,3,4
    4 DO 29 J4=1,J5
      I5=ID(J4)+1
      I6=ID(J4+1)
      IA1=IA(IM2)
      READ(2) (D(I),I=1,IA1)
      IM3=IM(IM2)
      IM2=IM2+1
      READ (4) (MD(I),I=1,IM3)
      DO 29 J6=I5,I6
      J7=0
      IF(J4.EQ.1) GO TO 79
      J7=M3(I5-1)+M2(I5-1)
   79 DO 29 L2=I3,I4
      L3=J6+M1(L2)-L2
      L4=I5+L3
      IF(L3) 29,50,50
   50 L5=M3(L2)+J6-L2-J2
      L7=M2(J6)
      DO 81 J=1,L7
      K=M4(6J)+J-1
      L6=MD(K)
      L14=M3(J6)+J-J7
      IF (L2-L6) 30,31,32
   30 L8=M2(L2)
      DO 33 L=1,L8
      L9=M4(L2)+L-1
      IF(MA(L9)-L6) 33,34,34
   33 CONTINUE
   34 K1=M3(L2)+L
      GOTO 35
   31 K1=M3(L2)
      GOTO 35
   32 K1=M3(L2)+L6-L2
   35 K1=K1-J2
      A(K1)=A(K1)-A(L5)*D(L14)
   81 CONTINUE
      B(L2)=B(L2) -A(L5)*B(J6)
   29 CONTINUE
    3 DO 49 L2=I3,I4
      K1=M3(L2)-J2
      IF(ABS(A(K1)).LT.EPS) GOTO 85
      B(L2)=B(L2)/A(K1)
      I13=L2+1
      DO 110 I=I13,I4
      IF(I-L2.GT.M1(I)) GOTO 110
      L6=M3(I)+L2-I-J2
      B(I)=B(I)-A(L6)*B(L2)
  110 CONTINUE
      L4=M2(L2)
      DO 49 L=1,L4
      L5=M3(L2)+L-J2
```

```
      A(L5)=A(L5)/A(K1)
      DO 49 I=I13,I4
      IF(I-L2.GT.M1(I)) GOTO 49
      L6=M3(I)+L2-I-J2
      L7=M4(L2)+L-1
      L8=MA(L7)
      IF(I-L8) 38,39,40
   39 L7=M3(I)
      GOTO 41
   40 L7=M3(I)+L8-I
      GOTO 41
   38 L9=M2(I)
      DO 42 J=1,L9
      L1=M4(I)+J-1
      IF(MD(L1)-(L8) 42,43,43
   42 CONTINUE
   43 L7=M3(I)+J
   41 L7=L7-J2
      A(L7)=A(L7)-A(L6)*A(L5)
   49 CONTINUE
      IF(ID(J1+1)-N) 46,47,47
   46 IA1=IA(IM2)
      WRITE(2) (A(I),I=1,IA1)
      J1=J1+1
      ENDFILE 2
      REWIND 2
      REWIND 4
      IM2=1
      GOTO 1
   47 ENDFILE 2
      IM3=IM(IM2)
      READ(4) (MD(I),I=1,IM3)
      GOTO 99
   69 CALL BACKSTEP(1H2)
      CALL BACKSTEP(1H2)
      CALL BACKSTEP(1H4)
      CALL BACKSTEP(1H4)
      IA1=IA(IM2)
      IM3=IM(IM2)
      READ(2) (A(I),I=1,IA1)
      READ(4) (MD(I),I=1,IM3)
      IM2=IM2-1
   99 L3=ID(J1)+1
      L4=ID(J1+1)
      L5=0
      IF(L3.LE.1) GO TO 76
      L5=M3(L3-1)+M2(L3-1)
   76 L1=M3(N)-L5
      IF (L9) 63,64,64
   64 L9=-1
      X(N)=B(N)
      L4=L4-1
   63 DO 68 I=L3,L4
      K1=L4-I+L3
      X(K1)=B(K1)
      S=0
      L8=M2(K1)
      L7=M4(K1)
      DO 68 J=1,L8
```

```
      L6=L7+J-1
      L1=M3(K1)+J-L5
      I1=MD(L6)
   68 X(K1)=X(K1)-A(L11)*X(I1)
      J1=J1-1
      IF(J1) 67,67,69
   85 PRINT 86,L2,I3,I4,K1,A(K1)
   86 FORMAT(10H SINGULAR:,4I5,E15.5)
   67 RETURN
      END
```

3/1. Program. Szalagmátrixú lineáris egyenletrendszer megoldása (lásd a 3-as feladat 3/1-es megoldása).

Az egyenletrendszer megoldását a SZALAG nevű szubrutin végzi, ennek hívása

CALL SZALAG (N, K, EPS),

ahol N az egyenletek száma, a szalag szélesség $2*K+1$. EPS vizsgálja a szingularitást, ha a pivotelem abszolút értékben kisebb mint EPS hibajelzés íródik ki és számolás félbe szakad. A szubrutinban a

COMMON /1/ A//B

utasításban levő A tömbben adjuk meg a szalagmátrix bal felső $(K+1)\times(K+1)$-es minorát, a B tömbben az egyenletrendszer jobboldalát. A szalagmátrix további részeit a 3-as feladat ismertetésénél leírt módon az 1-es file-on, disken adjuk meg. A megoldás a B tömbben képződik és kiíródik.

A számolás helyszükséglete: $N \leqq 7000$, $K \leqq 100$ méretű feladat számolásához (ha $N \cdot K \leqq 330\,000$) 89 qp operatív memória és 200 seg diszkterület.

A program listája:

```
      SUBROUTINE SZALAG(N,K,EPS)
C
C  THE SUBROUTINE SZALAG SOLVES A SYSTEM OF EQUATION WITH BAND MATRIX.
C      THE FIRST (K+1)*(K+1) PART OF BAND IS IN ARRAY A AT THE BEGINING
C      THEN ONE ROW AND COLUMN ARE TRANSFERRED FROM FILE 1 SUCCESSIVELY
C      AND IT WILL BE CONNECTED WITH MATRIX A.
C      AFTER EACH ELIMINATION STEP ONE ROW IS WRITTEN TO THE FILE 2.
C      THE BACK-SUBSTITUTION IS CARREID OUT WITH HELP OF FILE 2. THE RIGHT HAND
C      SIDE IS IN VECTOR B. ARRAYS A AND B ARE IN COMMON.
C  PARAMETER EPS LOOKS FOR SINGULARITY. IN CASE OF SINGULARITY A CERTAIN
C      MARK IS WRITTEN OUT .
C  PARAMETERS :  N IS THE NUMBER OF EQUATIONS;
C                K IS THE HALF WIDTH OF BAND.
C      THE SOLUTION IS FORMED IN ARRAY B.
C
      DIMENSION A(101,101),B(7000),C(201)
      COMMON /1/ A//B
      N1=N-1
      K1=K+1
      K2=K+K1
      REWIND 1
      DO 10 L=1, N1
      IF(ABS(A(1,1)).LT.EPS) GOTO 11
      A1=1/A(1,1)
      DO 1 J=2,K1
```

```
 1 A(1,J)=A1*A(1,J)
   B(L)=A1*B(L)
   DO 2 I=2,K1
   I1=L+I-1
   B(I1)=B(I1)-A(I,1)*B(L)
   DO 2 J=2,K1
 2 A(I,J)=A(I,J)-A(I,1)*A(1,J)
   WRITE(2) (A(1,J),J=2,K1)
   DO 3 I=1,K
   DO3 J=1,K
 3 A(I,J)=A(I+1,J+1)
   DO 4 I=1,K1
   A(K1,I)=0
 4 A(I,K1)=0
   IF(L-N+K) 5,10,10
 5 READ (1) (C(I),I=1,K2)
   DO 9 I=1, K
   A(K1,I)=C(I)
   J=K2-I+1
 9 A(I,K1)=C(J)
   A(K1,K1)=C(K1)
10 CONTINUE
   L=N
   IF(ABS(A(1,1)).LT.EPS) GO TO 11
   B(N)=B(N)/A(1,1)
   ENDFILE 2
   CALL BACKSTEP(1H2)
   DO 6 L=1,N1
   L1=N-L
   CALL BACKSTEP(1H2)
   READ(2) (A(1,I),I=1,K)
   CALL BACKSTEP(1H2)
   DO 6 I=1,K
   I1=L1+I
 6 B(L1)=B(L1)-A(1,I)*B(I1)
   GOTO 13
11 PRINT 12,L,A(1,1)
12 FORMAT(12H SZING,   :,I5,E12.5)
13 CONTINUE
   RETURN
   END
```

3/2. Program. Szalagmátrixú lineáris egyenletrendszer megoldására (lásd a 3-as feladat 3/2-es megoldása).

A program kártyáról olvassa a következő paramétereket:

N az egyenletek száma

K a fél szalagszélesség

EPS a szingularitást vizsgáló paraméter

IP = 1 esetén minden eliminációs lépés előtt

főelemet vizsgál és oszloponként kiválasztja a legnagyobb elemet, illetve végrehajtja a sorcserét; IP ≠ 1 esetén csak abban az esetben, ha a főelem abszolút értékben kisebb mint EPS.

A program a szalagmátrixot a 10-es file-ról, a jobboldalt a 20-as file-ról olvassa. A megoldás kiíródik. Szingularitás esetén hibajelzés adódik.

A számolás helyszükséglete: CDC gépen a program $N \leqq 16290$, $K \leqq 90$ (ha $N \cdot K \leqq 330\,000$) esetén hazsnálható, ehhez 89 qp operatív memória és 200 seg diszk terület szükséges.

A program listája:

```
C
C  THE PROGRAM SOLVES A SYSTEM OF EQUATIONS WITH BAND MATRIX.
C      THE COEFFICIENT BAND MATRIX IS IN FILE 10 BY ROWS-WISE (THE FIRST
C      AND LAST TRUNCATED ROWS ARE COMPLETED WITH ZEROS.)
C      THE RIGHT HAND SIDE IS IN FILE 20.
C      AFTER EACH ELIMINATION STEP ONE ROW IS WRITTEN TO FILE 2 AND ONE
C      COLUMN IS WRITTEN TO FILE 3. THE BACK-SUBSTITUTION IS CARRIED OUT
C      WITH HELP OF FILES 2 AND 3.
C  THE PROGRAM CHOOSES PIVOT ELEMENT BY COLUMNS AND THE ROWS ARE CHANGED
C      IF IT IS NECESSARY.
C  THE PARAMETER EPS LOOKS FOR SINGULARITY. IN CASE OF SINGULARITY A
C      CERTAIN MARK IS WRITTEN OUT.
C  PARAMETERS: N IS THE NUMBER OF EQUATION;
C      K IS THE HALF WIDTH OF BAND.
C  THE SOLUTION IS FORMED IN ARRAY B AND IT IS WRITTEN OUT.
C
      DIMENSION A(90,181),B(16290),C(181)
      EQUIVALENCE (A(1,1),B(1))
C
C  THE PARAMETERS ARE READ FROM CARD
C
      READ 11,K,N,IP,EPS
      K2=2*K+1
      K1=K+1
C
C  THE FIRST K ROWS OF BAND ARE READ FROM FILE 10.
C
      DO 1 I=1,K1
    1 READ(10,12) (A(I,J),J=1,K2)
      DO 2 I=1,K
      K3=K1+I-1
      DO 3 J=1,K3
      I1=K1-I+J
      J1=J+1
    3 A(I,J)=A(I,I1)
      DO 2 L=J1,K2
    2 A(I,L)=0.
      K4=K2-1
      K3=K1
      N1=N-1
      DO 4 L=1,N1
      IF(IP.EQ.1) GOTO 20
      IF1(ABS(A(1,1)).LT.EPS) GOTO 20
   27 A11./A(1,1)
      DO5 J= J=2,K3
    5 A(1,10 =A1*A(1,J)
      DO 10 JI=2,K1
      DO 10I=2,K3
   10 A(I,J)=A(I,J)-A(I,1)*A(1,J)
      WRITE(2) (A(1J),J=2,K2)
      WRITE(3) (A(J,1),J=1,K1)
```

```
      DO 6 M=1,K
      DO 6 J=1,K4
    6 A(M,J)=A(M+1,J+1)
      DO 7 I=1,K2
    7 A(K1,I)=0.
      DO 8 I=1,K1
    8 A(I,K2)=0.
      IF(L-N+K) 9,4,4
C
C  THE SUCESSIVE ROWS OF BAND ARE READ FROM FILE 10.
C
    9 READ(10,12) (A(K1,I),I=1,K2)
    4 CONTINUE
      IF(ABS(A(1,1)).LT.EPS) GOTO 25
      A1=1./A(1,1)
      ENDFILE 2
      ENDFILE 3
C
C  THE RIGHT HAND SIDE IS READ FROM FILE 20.
C
      READ(20,12) (B(I),I=1,N)
      REWIND 3
      DO 21 L=1,N1
      READ(3) (C(I),I=1,K1)
      B(L)=B(L)/C(1)
      DO 21 I=1,K
      I1=L+I
      IF(I1.(T.N) GOTO 21
      B(I1)=B(I1)-C(I+1)*B(L)
   21 CONTINUE
      CALL BACKSTEP(1H2)
      B(N)=B(N)*A1
      DO 23 L=2,N
      L1=N-L+1
      CALL BACKSTEP(1H2)
      READ(2) (C(I),I=2,K2)
      CALL BACKSTEP(1H2)
      DO 23 I=2,K2
      L2=L1+I-1
      B(L1)=B(L1)-C(I)*B(L2)
   23 CONTINUE
C
C  THE SOLUTION IS WRITTEN OUT.
C
      PRINT 13,(I,B(I),I=1,N)
      STOP
   20 M=1
      A1=ABS(A(1,1))
      DO 24 I=1,K
      IF(A1.GE.ABS(I,1)) GOTO 24
      M=I
      A1=ABS (A(I,1))
   24 CONTINUE
      K3=K2
      IF(M.EQ.1) GOTO 25
      DO 26 I=1,K2
      A2=A(1,I)
      A(1,I)=A(M,I)
   26 A(M,I)=A2
      GOTO 27
```

```
 25 PRINT 28,L,N,A1
 11 FORMAT(3I5,E10.5)
 12 FORMAT(5E16.8)
 13 FORMAT(1H0,I5,E15.7)
 28 FORMAT(14H SINGULAR :L=,2I5,E15.5)
    END
```

4. Program. Szimmetrikus szalagmátrixú lineáris egyenletrendszer megoldása (lásd 4-es feladat).

A programkártyáról beolvassa az N, NK, EPS paramétereket (2I5, F10.0) formátummal, ahol NK az egyenletek száma, N a fél szalagszélesség, EPS a szingularitást vizsgáló paraméter. Ha a *Cholesky módszer* alkalmazása közben a pivot abszolút értékben kisebb lesz mint EPS hibajelzés íródik ki és a számolás félbeszakad

A program a szimmetrikus szalagmátrix főátlótól jobbra eső felét az 1-es diszk file-ről olvassa, elvégzi *Cholesky módszerrel* az $\mathbf{A}=\mathbf{LR}$ faktorizálást. Ezután beolvassa a 2-es diskfile-ről az egyenletrendszer jobboldalát és a (2.4) rendszerek megoldásával kiszámítja az egyenletrendszer megoldását. Az eredmény a B tömbben képződik és kinyomtatódik.

A számolás helyszükséglete: 83 qp operatív memória és 200 seg diszk memória felhasználásával $NK \leqq 16\,110$, $N \leqq 179$ méretű feladat számolható (de $N \cdot NK \leqq 10^6$).

A program listája:

```
C
C  THE PROGRAM SOLVES A SYSTEM OF LINEAR EQUATION WITH
C  SYMMETRIC BAND MATRIX OF MEAN OF CHOLEVSKY METHOD
C
      DIMENSION A(16110),B(16110),M(179),C(179)
      EQUIVALENCE (A(1),B(1))
      COMMON /10/ A
      READ 22,NK,N,EPS
   22 FORMAT(2I5,F10.0)
      NN=N*(N+1)/2
      REWIND 1
      REWIND 3
      N2=N+1
      N3=NK+N
      M(1)=0
      M1=N
      DO 1 K=2,N
      M(K)=M(K-1)+M1
    1 M1=M1-1
      DO 2 I=1,NN
    2 A(I)=0
      DO 10 KK=1,NK
      READ(1) (A(I),I=1,N)
      N1=N-1
      S=A(1)
      DO 3 I=2,N
      M1=M(I)+1
    3 S=S-A(M1)*X(M1)
      S=SQRT(S)
      A(1)=S
      IF(S.LT.EPS) GOTO 9
      DO 4 I=2,N
```

```
      I1=N-I+1
      S1=A(I)
      DO 5 J=2,I1
      M1=M(J)+1
      M2=M(J)+I
    5 S1=S1-A(M1)*A(M2)
    4 A(I)=S1/S
      DO 6 I=1,N
      M1=M(I)+1
    6 C(I)=A(M1)
      WRITE(2) (C(I),I=1,N)
      DO 7 K1=1,N1
      DO 7 L=1,K1
      K=N-K1
      I=M(K)+L+1
      J=M(K+1)+L
    7 A(J)=A(I)
   10 CONTINUE
      ENDFILE 2
      DO 8 I=1,N
    8 B(I)=0
      READ(3) (B(I),I=N2,N3)
      REWIND 2
      DO 11 K=1,NK
      READ(2) (C(I),I=1,N)
      K1=N+K
      S=B(KA)
      DO 12 I=2,N
      K2=K1-I+1
   12 S=S-B(K2)*C(I)
   11 B(K1)=S/C(1)
      DO 13 K=1,NK
      CALL BACKSTEP(1H2)
      READ(2) (C(I),I=1,N)
      K1=N3-K+1
      CALL BACKSTEP(1H2)
      B(K1)=B(K1)/C(1)
      DO 14 I=1,N1
      K2=K1-I
   14 B(K2)=B(K2)-B(K1)*C(I+1)
   13 CONTINUE
      PRINT 16,N,NK
   16 FORMAT(12H A MEGOLDÁS:,/2I5//)
      PRINT 17,(B(I),I=N2,N3)
   17 FORMAT(10=12.5)
      GOTO 18
      9 PRINT 19,KK.S
   19 FORMAT(13H SZINGULARIS:,I5,E15.5)
   18 CONTINUE
      END
```

5. Program. Szimmetrikus ritka mátrix sajátértékeinek kiszámítása (lásd 5-ös feladat).

A program beolvassa az N, az EPS és az EPS2 paramétereket kártyáról (I4, 2E16.9)-es formátummal, ahol N a mátrix rendszáma EPS a sajátértékekre vonatkozó pontossági igény, EPS2 vizsgálja a szingularitást. Ezután olvassa a 2-es file-ről az IS tömböt (15I5-ös formátummal) az 1-es file-ről pedig a szimmetrikus

mátrix főátlótól jobbra levő nem zérus elemeit és azok oszlopindexeit (4(I4, E16.9))-es formátumban. Az 1-es file-on a nem zérus elemeket sorfolytonosan kell rendezni. Minden sor után egy 0 oszlopindex írandó. A nem zéró elemek végét egy negatív oszlopindex jelzi.

A program a *szimmetrikus Lanczos módszert* számolja, amihez felhasználja az AV (N, V, S) szubrutint, ami az $S=AV$ mátrixvektor szorzást végzi. A *Lánczos módszer* a szimmetrikus három átlós mátrixot az E(.), D(.) tömbökben helyezi el majd a QR(N, SMACHEPS, E, D, IER) szubrutin számítja az *implicit QR módszert*. A sajátértékek a D(.) tömbben képződnek és kinyomtatódnak azok hibáival együtt. A v_k ortonormált vektorrendszer ortogonalitásának javítását az ORTSIM szubrutin végzi. Itt történik annak megvizsgálása is, hogy a v_k vektor abszolút értéke nem lett-e nagyon kicsi, kisebb mint EPS2. Ebben az esetben új v_k vektort keres a szubrutin.

A számolás helyszükséglete 65 qp a gyorsmemóriában és 65 seg a diszk-en. Ilyen helyfoglalás mellett $N \leqq 500$ és a főátló feletti nem zéró elemek száma $\leqq 5000$.

A program listája:

```
C
C  THE PROGRAM COMPUTS THE EIGENVALUES OF A SYMMETRIC SPARSE MATRIX.
C  THE SYMMETRIC LACZOS METHOD THEN THE IMPLICIT QR METHOD ARE USED.
C
      DIMENSION C(500),B(500),V(500),W(500),S(500),I1(5),A1(5)
      COMMON /10/ M(5000),A(5000),IS(500)
      REWIND 1
      REWIND 2
C
C  THE INPUT OF PARAMETERS:
C  N — IS THE ORDER OF MATRIX
C  EPS — IS THE ACCURACY FOR EIGENVALUES
C  EPS2 INVESTIGATES THE SINGULARITY
C
      READ 17,N,EPS,EPS2
      PRINT 17,N,EPS,EPS2
      N1=N+1
C
C  THE ARRAY ELEMENT IS(I) LOOKS FOR THE FIRST NONZERO ELEMENT IN ROW I.
C
      READ(2,16) (IS(I),I=1,N1)
      K=1
C
C  THE NONZERO ELEMENTS AND THEIR COLUMN INDEXES ARE READ FROM FILE 1
C
    1 READ(1,15) (I1(I),A1(I),I=1,4)
      DO 2 I=1,4
      IF(I1(I)) 10,2,4
    4 M(K)=I1(I)
      A(K)=A1(I)
      K=K+1
    2 CONTINUE
      GOTO 1
   10 DO 5 K=1,N
    5 V(K)=0.
      PRINT 16,(IS(I),I=1,N1)
      NN1=IS(N+1)-1
```

```
      PRINT 15,(M(I),A(I),I=1,NN1)
      V(1)=1
      V(N1)=1.
      WRITE(3) (V(I),I=1,N1)
      C(1)=0
      DO 6 K=1,N
      CALL AV(N,V,S)
      DO 7 L=1,N
    7 B(K)=B(K)+V(L)*S(L)
      DO 8 L=1,N
    8 W(L)=S(L)-B(K)*V(L)-C(K)*W(L)
      CALL ORTSIM(N,K,W,S,C,EPS2,T)
      DO 11 L=1,N
      D=V(L)
      V(L)=W(L)/T
   11 W(L)=D
    6 CONTINUE
      PRINT 13,(I,C(I),B(I),I=1,N1)
      CALL QR(N,EPS,C,B,IER)
C
C  THE EIGENVALUES END THEIR ACCURACY ARE WRITTEN OUT
C
      PRINT 13,(I,C(I),B(I),I=1,N)
   13 FORMAT(1H0,I5,2E18.9)
   15 FORMAT(4(I4,E16.9))
   16 FORMAT (15I5)
   17 FORMAT(I4,2E16.9)
      END
      SUBROUTINE QR(N,SMACHEPS,E,D,IER)
      DIMENSION E(1),D(1)
C
C  THE SUBROUTINE QR COMPUTES THE EIGENVALUES BY MEAN OF QR METHOD FOR THE
C  THREE DIAGONAL MATRIX.
C
      IER=0
      DO 1000 I=2,N
 1000 E(I-1)=E(I)
      E(N)=0
      K=N-1
      DO 1001 L=1,N
      J=0
 1002 DO 1003 M=L,K
      IF(ABS(E(M)).LE.SMACHEPS*(ABS(D(M))+ABS(D(M+1)))) GOTO 1004
 1003 CONTINUE
      M=N
 1004 P=D(L)
      IF(M.EQ.L) GOTO 1001
      IF(J.EQ.30) GOTO 1005
      J=J+1
      G=(D(L+1)-P)/(2*E(L))
      R=SQRT(1.+G*G)
      Q=1
      IF(G.LT.0.) Q=-1.
      G=D(M)-P+E(L)/(G+Q*R)
      S=1.
      C=1.
      P=D(M)
      MM=M-1
      DO 1006 II=L,MM
      I=M-1+L-II
```

```
      F=S*E(I)
      B=C*E(I)
      IF(ABS(F).LT.ABS(G)) GOTO 1007
      C=G/F
      R=SQRT(1.+C*C)
      E(I+1)=F*R
      S=1./R
      C=C/R
      GOTO 1008
 1007 C=F/G
      R=SQRT(1.+C*C)
      E(I+1)=G*R
      S=C/R
      C=1./R
 1008 F=C*D(I)-S*B
      G=C*B-S*P
      R=D(I)+P
      P=C*F-S*G
      G=S*F+C*G
      D(I+1)=R-P
 1006 CONTINUE
      D(L)=P
      E(L)=G
      E(M)=0
      GOTO 1002
 1001 CONTINUE
      DO 1009 I=1,N
      K=I
      P=D(I)
      III=I+1
      DO 1010 J=III,N
      IF(D(J).GE.P) GOTO 1011
      K=J
 1010 P=D(J)
 1011 IF(K.EQ.I) GOTO 1012
      D(K)=D(I)
      D(I)=P
 1012 CONTINUE
 1009 CONTINUE
      RETURN
 1005 IER=1
      RETURN
      END
      SUBROUTINE AV(N,V,S)
      DIMENSION V(1),S(1)
      COMMON/10/ M(5000),A(5000),IS(500)
C
C  THE SUBROUTINE AV COMPUTES THE PRODUCT S=A*V.
C
      DO 1 K=1,N
      S(K)=0
      K1=IS(K)
      K2=IS(K+1)-1
      DO 2 I=K1,K2
      J=M(I)
    2 S(K)=S(K)+A(I)*V(J)
      K3=K-1
      DO 3 I=1,K3
      K1=IS(I)
      K2=IS(I+1)-1
```

```
      DO 4 J=K1,K2
      IF(M(J).EQ.K) GOTO 5
    4 CONTINUE
    3 CONTINUE
      GOTO 1
    5 S(K)=S(K)+A(J)*V(I)
      GOTO 3
    1 CONTINUE
      RETURN
      END
      SUBROUTINE ORTSIM(N,K,W,S,C,EPS,T)
      DIMENSION W(1),S(1),C(1)
C
C  THE SUBROUTINE ORTSIM PERFORMS THE RE-ORTHOGONALIZATION OF THE MATRIX
C
      REWIND 3
      DO 1 L=1,K
      Z1=0
      N1=N+1
      READ(3) (S(J),J=1,N1)
      DO 2 I=1,N
    2 Z1=Z1+S(I)*W(I)
      Z2=Z1/S(N1)
      DO 3 I=1,N
    3 W(I)=W(I)-Z2*S(I)
    1 CONTINUE
      F=0
      DO 18 L=1,N
   18 F=F+W(L)*W(L)
      T=SQRT(F)
      C(K+1)=T
      IF(T.GE.EPS) GOTO 7
      IF(K.EQ.N) GOTO 7
C
C  A NEW ORTHOGONAL VECTOR IS GENERATED , IF W(.) IS SMALL
C
      DO 11 L=1,N
      DO 12 I=1,N
   12 W(I)=0
      W(L)=1.
      REWIND 3
      DO 13 I=1,K
      READ(3) (S(J),J=1,N1)
      U=0
      DO 4 J=1,N
      U=U+S(J)*W(J)
    4 CONTINUE
      Z=U/S(N1)
      DO 5 J=1,N
    5 W(J)=W(J)-Z*S(J)
   13 CONTINUE
      F=0
      DO 6 I=1,N
    6 F=F+W(I)*W(I)
      T=SQRT(F)
      C(K+1)=0
      PRINT 9,N,K,L,T,EPS,Z
      IF(T.GE.EPS) GOTO 7
   11 CONTINUE
      PRINT 8,K,EPS,T
```

```
8 FORMAT (13H W EQUAL ZERO,I5,2E15.5)
  STOP
7 CONTINUE
  W(N1)=F
  WRITE(3) (W(I),I=1,N1)
9 FORMAT(3I5,3E15.5)
  RETURN
  END
```

6. Program. Nem szimmetrikus ritka mátrix sajátértékeinek kiszámítása (lásd 6-os feladat).

A program beolvassa kártyáról az N, R1, R2, EPS, EPS1 paramétereket (I5, 4F15.8) formátummal, ahol N a mátrix rendszáma, EPS a sajátértékekre vonatkozó pontossági igény, EPS1 vizsgálja a szingularitást. A program az (R1, R2) intervallumba eső összes valós sajátértéket számítja. A mátrix nem zéró elemeit és azok oszlopindexeit az 1-es file-ről olvassa (4(I4, E16.9)) formátummal. A nem zéró elemeket sorfolytonosan tömören kell elhelyezni az 1-es file-on. Minden sor után egy 0 oszlopindex írandó. Egy negatív oszlopindex jelzi az adatok végét. A program a *Lánczos módszert* számítja. Ehhez felhasználja az AX(N, S, X) és az AY(N, S, Y) szubrutinokat, amelyek az $S=AX$, illetve $S=A^TY$ mátrix vektor szorzásokat végzik. A *Lánczos módszerekből* nyert háromátlós mátrix-ot a B(.) és a C(.) tömbben helyezi el (a harmadik átló 1-esekből áll). Ezután a SEAT (N, R1, R2, I9, S, EPS) és a SAJS2(N, P, NV) szubrutinok számolják ki az (R1, R2) intervallumba eső sajátértékeket Bisection módszerrel. A sajátértékek ki is nyomtatódnak. Az ORTOG (N, K, X2, Y, EPS, M) szubrutin javítja az $\mathbf{X}_k$ és $\mathbf{Y}_k$ vektor rendszer biortogonalitását, majd megvizsgálja, hogy az $\|\mathbf{X}_k\| \geqq$EPS1 és $\|\mathbf{Y}_k\| \geqq$EPS1 teljesül-e? Ha az teljesül, akkor azt is megvizsgálja, hogy a biortogonális $\mathbf{X}_k$, $\mathbf{Y}_k$ vektorrendszerre valamilyen k esetén $|(\mathbf{Y}_k^T, \mathbf{X}_k)| \geqq$EPS1 teljesül-e? A *Lánczos módszer* alkalmazásához ennek teljesülnie kell. Ha a fenti egyenlőtlenségek valamelyike nem teljesül, akkor új $\mathbf{X}_k$ és $\mathbf{Y}_k$ vektorokat generál, amelyek biortogonálisak az előzőekre. Ezután folytatja a számolást.

A program 71 qp operatív memória és 115 seg diszk memória felhasználásával alkalmas $N \leqq 500$ méretű feladat megoldására, ha a nem zéró elemek száma $\leqq 5000$.

A program listája:

```
C
C  THE PROGRAM COMPUTES THE REAL EIGENVALUES BELONG IN INTERVAL (R1,R2)
C  THE LANCZOS METHOD THEN THE BISECTION METHOD ARE USED.
C
C  THE INPUT OF PARAMETERS:
C  N — IS THE ORDER OF MATRIX
C  EPS — IS THE ACCURACY FOR EIGENVALUES
C
      DIMENSION IS(500),IO(5000),A(5000),B(500),C(500),I1(4),A1(4),X1(50
     10),X2(500),Y1(500),Y2(500),S(500)
      COMMON /1/ B,C /2/ IS,IO,A
      READ 1,N,R1,R2,EPS,EPS1
      PRINT 1,N,R1,R2,EPS,EPS1
      N1=N+1
      N2=N-1
      REWIND 1
      REWIND 2
      READ(1,9) (IS(I),I=1,N1)
      I2=IS(N+1)-1
```

```
C
C  THE NONZERO ELEMENTS AND THEIR COLUMN INDEXES ARE READ FROM FILE 2
C  THE ARRAY ELEMENT IS(I) LOOKS FOR THE FIRST NONZERO ELEMENT IN ROW I.
C
      READ(2,10) (IO(I),A(I),I=1,I2)
      PRINT (IO(I),A(I),I=1,I2)
      DO 50 L=1,N
      DO 11 I=1,N
      Y1(I)=0.
      X1(I)=0.
      X2(I)=0
      Y2(I)=0
   11 CONTINUE
      X1(L)=1.
      Y1(L)=1.
      X1(N1)=1.
      WRITE(3) (X1(I),I=1,N1)
      WRITE(4) (Y1(I),I=1,N)
      C(1)=0
      S1=0
      DO 12 I=1,N
   12 S1=S1+X1(I)*Y1(I)
      DO 19 K=1,N
      CALL AX(N,S,X1)
      S2=0
      DO 13 I=1,N
   13 S2=S2+Y1(I)*S(I)
      IF(ABS(S1).LT.EPS1) GOTO 50
      PRINT 25,S1,S2
      B(K)=S2/S1
      DO 14 I=1,N
   14 X2(I)=S(I)-B(K)*X1(I)-C(K)*X2(I)
      CALL AY(N,S,Y1)
      DO 15 I=1,N
   15 Y2(I)=S(I)-B(K)*Y1(I)-C(K)*Y2(I)
      CALL ORTOG(N,K,X2,Y2,EPS1,M,S3)
      C(K+1)=S3/S1
      S1=S3
      IF(M.LT.O) C(K+1)=0
      DO 17 I=1,N
      E=X1(I)
      X1(I)=X2(I)
      X2(I)=E
      E=Y1(I)
      Y1(I)=Y2(I)
   17 Y2(I)=E
   19 CONTINUE
      IF(L.EQ.N) GOTO 20
      GOTO 51
   50 PRINT 10,L,S1,K
   51 CONTINUE
      PRINT 25,(B(I),C(I),I=1,N1)
      DO 18 I=1,N
   18 C(I)=C(I+1)
      CALL SEAT(N,R1,R2,NS,S,EPS)
C
C  THE EIGENVALUES ARE WRITTEN OUT
C
      PRINT 22,NS,(S(I),I=1,NS)
      GOTO 24
```

```
   20 PRINT 23,L,S1
   24 CONTINUE
   25 FORMAT(2E18.9)
    9 FORMAT (1515)
   10 FORMAT(4(I4,E16.9))
   23 FORMAT(7H SZING:,I5,E15.5)
   22 FORMAT(I5/,(5E18.8))
    1 FORMAT(I5,4E15.8)
      END
      SUBROUTINE AX(N,S,X)
      DIMENSION S(1),X(1)
      COMMON /2/ IS(500),IO(5000),A(5000)
C
C  THE SUBROUTINE AX COMPUTES THE PRODUCT S=A*X.
C
      DO 1 I=1,N
      S(I)=0
      KK=IS(I)
      KV=IS(I+1)-1
      DO 1 K=KK,KV
      J=IO(K)
    1 S(I)=S(I)+A(K)*X(J)
      RETURN
      END
      SUBROUTINE AY(N,S,Y)
      DIMENSION S(1),Y(1)
      COMMON /2/ IS(500),IO(5000),A(5000)
C
C  THE SUBROUTINE AY COMPUTES THE PRODUCT S=Y*A
C
      DO 1 J=1,N
      S(J)=0
      DO 1 I=1,N
      IK=IS(I)
      IV=IS(I+1)-1
      DO 3 K=IK,IV
      IF(IO(K).EQ.J) GOTO 4
    3 CONTINUE
    1 CONTINUE
      GOTO 5
    4 S(J)=S(J)+A(K)*Y(I)
      GOTO 1
    5 RETURN
      END
      SUBROUTINE SEAT(N,R1,R2,I9,S,EPS)
      DIMENSION S(1)
C
C  THE SUBROUTINE SEAT COMPUTES THE BISECTION METHOD
C
      NS=1
      CALL SAJS2(N,R1,N1)
      CALL SAJS2(N,R2,N2)
      PRINT 13,N,N1,N2,R1,R2
      IF(N2-N1) 1,1,2
C
C  A MARC IS WRITTEN OUT IF EIGENVALUES ARE NOT IN INTERVAL (R1,R2)
C
    1 PRINT 3,R1,R2
      GOTO 10
    2 I9=N2-N1
```

```
C
C  IT IS WRITTEN OUT HOW MANY EIGENVALUES ARE IN INTERVAL (R1,R2)
C
      PRINT 4,I9,R1,R2
    5 CONTINUE
      IF(R2-R1-EPS.LE.0.) GOTO 20
      P=(R1+R2)/2
      CALL SAJS2(N,P,N3)
      IF(N2-N3) 7,7,8
    7 R2=P
      GOTO 5
    8 IF(N3-N1) 21,21,22
   21 R1=P
      GOTO 5
   22 R3=P
   23 P=(R3+R2)/2
      IF(R2-R3-EPS.LE.0.0) GOTO 20
      CALL SAJS2(N,P,N4)
      IF(N2-N4) 19,19,18
   19 R2=P
      GOTO 23
   18 R3=P
      GOTO 23
   20 S(NS)=P
      IF(NS.EQ.I9) GOTO 10
      NS=NS+1
      R2=P-2*EPS
      N2=N2-1
      GOTO 5
   10 CONTINUE
   13 FORMAT(3I5,2E16.6)
    3 FORMAT(23H IS NOT EIGENVALUE IN (,2E15.6,2H))
    4 FORMAT(I5,S1H EIGENVALUES ARE IN (,2E15.6,2H))
      RETURN
      END
      SUBROUTINE SAJS2(N,P,NV)
      COMMON /1/ B(500),C(500)
      LOGICAL L
C
C  SUBROUTINE SAJS2 COMPUTES THE NUMBER OF CHANGE OF SIGN IN THE SEQUENCE
C  OF MINORS.
C
      NV=0
      L=.TRUE.
      D1=1.
      D2=B(1)-P
      IF(D2) 3,5,5
    3 L=.NOR.L
      NV=1
    5 CONTINUE
      DO 1 I=2,N
      D=D2*(B(I)-P)-C(I-1)*D1
      D1=D2
      D2=D
      IF(L.AND.D.GE.0.0.OR..NOT.L.AND.D.LT.0.0.)  GOTO  1
      NV=NV+1
      L=.NOT.L
    1 CONTINUE
      PRINT 2,NV,P,D
    2 FORMAT (I5,2E16.7)
```

```
      RETURN
      END
      SUBROUTINE ORTOG(N,K,X2,Y2,EPS,M,S3)
      DIMENSION X2(1),Y2(1),U(500),V(500)
C
C  THE SUBROUTINE ORTOG PERFORMS THE RE-BIORTHOGONALIZATION OF THE MATRIX
C
      REWIND 3
      REWIND 4
      M=0
      DO 1 L=1,K
      Z1=0
      Z2=0
      N1=N+1
      READ(3) (U(J),J=1,N1)
      READ(4) (V(J),J=1,N)
      DO 2 I=1,N
      Z1=Z1+V(I)*X2(I)
      Z2=Z2+U(I)*Y2(I)
    2 CONTINUE
      Z5=Z1/U(N1)
      Z6=Z2/U(N1)
      DO 3 I=1,N
      X2(I)=X2(I)-Z5*U(I)
    3 Y2(I)=Y2(I)-Z6*V(I)
    1 CONTINUE
      DO 17 L=1,N
      S1=0
      S2=0
      DO 4 I=1,N
      S1=S1+X2(I)*X2(I)
    4 S2=S2+Y2(I)*Y2(I)
C
C  THE SUBROUTINE GENERATES A NEW BIORTOGONAL VECTORS
C  IF X2 OR Y2 IN NORM LES THEN EPS.
C
      IF(S1.GE.EPS.AND.S2.GE.EPS) GOTO 15
      DO 6 I=1,N
    6 X2(I)=0
      X2(L)=1.
      REWIND 3
      REWIND 4
      DO 7 LL=1,K
      READ(3) (J(I),I=1,N1)
      READ(4) (V(I),I=1,N)
      Z1=0
      DO 8 I=1,N
      Z1=Z1+V(I)*X2(I)
    8 CONTINUE
      Z3=Z1/U(N1)
      DO 9 I=1,N
    9 X2(I)=X2(I)-Z3*V(I)
    7 CONTINUE
      DO 16 I=1,N
   16 Y2(I)=0.
      Y2(L)=1.
      REWIND 3
      REWIND 4
      DO 17 LL=1,K
      READ(3) (J(I),I=1,N1)
```

```
      READ(4) (V(I),I=1,N)
      Z1=0
      DO 18 I=1,N
      Z1=Z1+U(I)*Y2(I)
   18 CONTINUE
      Z3=Z1/U(NL)
      DO 19 I=1,N
   19 Y2(I)=Y2(I)-Z3*U(I)
      M=-1
   17 CONTINUE
   15 CONTINUE
      S3=0
      DO 20 I=1,N
   20 S3=S3+X2(I)*Y2(I)
      X2(N1)=S3
      WRITE(3) (X2(I),I=1,N1)
      WRITE(4) (Y2(I),I=1,N)
      RETURN
      END
```

IRODALOM

[1] WILKINSON, J. H., *The Algebraic Eigenvalue Problem,* (Clarendon Press, Oxford, 1965).

[2] GERGELY, J., „Nagyméretű mátrixok invertálása", *Alkalmazott Matematikai Lapok* 4 (1978) 143—149.

[3] WILLOUGHBY, R. A., Proceedings of the symposium on sparse matrices and their application, IBM Watson Research Center, 1968.

[4] REID, J. K., Large sparse sets of linear equations, Proceedings of the Oxford conference, Academic Press, London, 1971.

[5] TEWARSON, R. P., *Sparse Matrices* (Academic Press, New York and London, 1973).

[6] DUFF, I. S., "A survey of sparse matrix research", *Proceedings of the IEEE* **65** (1977).

[7] REID, J. K., *FORTRAN Subroutines for Handling Sparse Linear Programming Bases* (Oxfordshire, 1976).

[8] GERGELY, J., „Numerikus módszerek sparse mátrixokra", *MTA SZTAKI Tanulmányok* **26** (1974).

[9] DOLD, A. and ECHMANN, B., *Sparse Matrix Techniques* (Springer, Copenhagen, 1976).

[10] PETERS, G. and WILKINSON, J. H., "Eigenvalues of $\mathbf{Ax}=\lambda\mathbf{Bx}$ with band symmetric **A** and **B**", *Computing Journal* **12** 398—404.

(Beérkezett: 1980. június 16.)

DR. GERGELY JÓZSEF
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST, ÚRI U. 49.

METHODS AND PROGRAMS FOR SPARSE MATRICES

J. GERGELY

In the paper methods and programs are investigated for the solution of six problems of linear algebra where the matrices of the problems are sparse ones. Linear systems of equation are solved in the first four programs, the eigenvalues are computed in the 5-th and 6-th program in the symmetric, nonsymmetric or band cases. The inverse and determinant also are computed in the first program.

A kiadásért felel az Akadémiai Kiadó igazgatója
Műszaki szerkesztő: Sándor István
A kézirat nyomdába érkezett: 1981. június 3. — Terjedelem: 20,12 (A/5 ív)
81-2867 — 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 kell beküldeni.

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: 50,— Ft

INDEX: 26 498
ISSN 0133—3399

TARTALOMJEGYZÉK

INDEX

ALKALMAZOTT MATEMATIKAI LAPOK

A MAGYAR TUDOMÁNYOS AKADÉMIA MATEMATIKAI ÉS FIZIKAI TUDOMÁNYOK OSZTÁLYÁNAK KÖZLEMÉNYEI

VI. KÖTET

AKADÉMIAI KIADÓ, BUDAPEST
1980

TARTALOMJEGYZÉK

A külföldi szakirodalomból

INDEX

From the foreign literature

81-2867 — Szegedi Nyomda — Felelős vezető: Dobó József igazgató