Alkalmazott matematikai lapok

1981/1-2

A MAGYAR TUDOMÁNYOS AKADÉMIA
MATEMATIKAI ÉS FIZIKAI TUDOMÁNYOK
OSZTÁLYÁNAK KÖZLEMÉNYEI

7.

KÖTET

AKADÉMIAI KIADÓ, BUDAPEST

ALKALMAZOTT MATEMATIKAI LAPOK

A MAGYAR TUDOMÁNYOS AKADÉMIA MATEMATIKAI ÉS FIZIKAI TUDOMÁNYOK OSZTÁLYÁNAK KÖZLEMÉNYEI

VII. 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.

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 7 (1981) 1—71

ADAPTÍV ELEMEK A LINEÁRIS PROGRAMOZÁSBAN, II.

MAROS ISTVÁN

Budapest

A számítógépes LP rendszerek fejlesztése jelenleg elsősorban azt célozza, hogy minél nagyobb méretű LP feladatokat minél olcsóbban, biztonságosabban és kényelmesebben lehessen megoldani. Az erre szolgáló rengeteg felhasználói paraméter közti eligazodás általában nagyon nehéz, és a paraméterek értékének a helyes megválasztása még egy adott feladattípusra is általában csak hosszabb gyakorlattal, drága tanulópénz árán sajátítható el. A cikk foglalkozik azzal, hogy valójában mit is jelent a paraméterek helyes megválasztása, ebben milyen szerep juthat az adaptivitás elve alkalmazásának, majd bemutat néhány új adaptív elemet. Beszámol az új elemekkel szerzett számítástechnikai tapasztalatokról is.

1. Bevezetés

A lineáris programozás, mint matematikai eszköz, hazánkban is egyre fontosabb szerepet játszik a gazdasági és műszaki gyakorlatban egyaránt. Ezt — a lineáris programozásban rejlő potenciális előnyök fokozatos felismerésén túlmenően — az teszi lehetővé, hogy az LP feladatok megoldásának elengedhetetlen segédeszközei, a számítógépek egyre szélesebb körben terjednek el. Természetesen a számítógépek is csak akkor tudnak megoldani LP feladatokat, ha fel vannak szerelve olyan programmal (programcsomaggal), amelyik képes az LP feladatok megoldására. A lineáris programozás elméletének megjelenésével szinte egyidőben megtörténtek az első lépések gépi megoldó algoritmusok és programok kidolgozására. Ezek a programok a DANTZIG [5] által kidolgozott szimplex módszer első számítógépes implementációinak tekinthetők. Képességeik — mai szemmel nézve — szerények voltak. Segítségükkel lényegében véve iskolapélda méretű feladatokat lehetett megoldani. Ennek egyik fő oka az volt, hogy maguk a rendelkezésre álló számítógépek szerény kapacitásúak voltak: memóriájuk és műveleti sebességük egyaránt kicsi volt. A kis memória azt eredményezte, hogy csak kevés adat és kisebb méretű, kevéssé bonyolult programok fértek el benne. A kis műveleti sebesség pedig eleve akadályt jelentett minden nagy műveletigényű számítási munka számára.

A számítógépek teljesítménye rohamos növekedésének és a gyakorlatban felmerülő egyre nagyobb LP feladatoknak a hatására az LP rendszerek is jelentős fejlődésnek indultak. A konkrét feladatok megoldása során szerzett tapasztalatok termékenyítő hatással voltak a szimplex módszer elméletére, és a szép számmal megjelenő új meggondolások szinte kivétel nélkül a gyakorlatban felmerült problémák megoldására való törekvés kapcsán születtek.

A sokirányú fejlődés eredményeképpen néhány számítógéptípuson ma már olyan LP programcsomagok léteznek, melyek több ezer feltételből és több tízezer változóból álló feladatokat is sikerrel oldottak meg. A még kedvezőbb tulajdonságú LP

rendszerek kidolgozására irányuló kutató-fejlesztő munka azonban egyáltalán nem állt meg. Ennek több oka van. Egyrészt változtak az LP rendszerekkel szembeni követelmények. Amikor a számítógép használatának költsége az igénybevett erőforrások számától és nagyságától függ, nem előnyös, ha az eljárás nagyon memóriaigényes, vagy nagyméretű mágneslemez területekre van szükség, illetve túl gyakran kell a háttértárolóval adatkommunikációt folytatnia. Hasonlóképpen lényeges szempont lett, hogy az LP rendszereknek növekedjék a használati kényelme, ne csak gyakorlott specialisták tudjanak dolgozni velük. Másrészt jelenleg még az a helyzet, hogy nem sikerült a szimplex módszernek olyan variánsát kidolgozni, mely minden józan igénynek eleget tudna tenni, vagy amit az ésszerűség határain belül a lehető legjobbnak nevezhetnénk. A most létező LP rendszerekre ugyanis az jellemző, hogy ha például egy bizonyos szempontból az egyik jobb a másiknál, egy másik szempontból fordított lehet a rangsor. A helyzet aztán tovább bonyolódik, ha több szempontot veszünk figyelembe és több LP rendszert hasonlítunk össze.

A mai kutatómunka elsősorban azt célozza, hogy minél nagyobb méretű LP feladatokat minél olcsóbban, biztonságosabban és kényelmesebben lehessen megoldani. Az ennek érdekében már eddig is kifejlesztett eljárások általában több szabad paramétert tartalmaznak, melyek helyes megválasztása nagy mértékben segíti a kitűzött cél elérését. A jelenlegi korszerű LP programcsomagoknál a felhasználói paraméterek száma 50—100 körül van. A paraméterek kedvező értéke feladatról feladatra, sőt, az iterációk során akár lépésről lépésre is változhat, és előre általában nem lehet tudni, hogy milyen választás a célravezető. Első hallásra ez elég kellemetlen jelenségnek látszik. Méginkább annak tűnik, ha hozzátesszük, hogy a paraméterek különböző értékeire nem pusztán gyorsabban vagy lassabban működik az eljárás, hanem akár helytelen választ is adhat a problémára (például lehetséges megoldás létezése esetén azt állapítja meg, hogy nincs lehetséges megoldás), sőt az is előfordulhat, hogy egyáltalán nem kapunk választ (például két invertálás közötti javítást a második invertálás nem reprodukálja, emiatt ciklikusan javítás és visszaesés következik egymás után). Ilyen körülmények közt felmerül a kérdés: mi garantálja az LP programcsomagok megbízhatóságát, illetve az LP-ben kevéssé jártas felhasználónak mekkora esélye van arra, hogy problémáját meg tudja oldani? A riasztó kérdésre viszonylag elfogadható választ lehet adni. A programcsomagokban majdnem minden paraméternek van előre beállított standard értéke, méghozzá olyan, mely az átlagos típusú feladatok esetén — a készítők tapasztalata szerint — jól látja el szerepét. A felhasználónak csak abban az esetben kell valamilyen értéket megadnia, ha a standard érték nem látszik megfelelőnek.

A sok paraméter szerepe világos: segítségükkel lehet a programot minél jobban „hozzáidomítani" az aktuális feladathoz, hogy azt valamilyen szempontból a legkedvezőbben (pl. legolcsóbban, leggyorsabban, vagy legpontosabban) oldja meg. Ez a lehetőség azonban a helytelen választás veszélyét is magában hordozza, s nem kevés azon esetek száma, amikor gyakorlott felhasználó kezében is sok „üresjáratot" végzett egy-egy korszerű LP programcsomag. Tekintettel arra, hogy az ilyen eseményekről indokolatlanul kevés az irodalmi beszámoló, ezért a szakmai közvélemény egy részének a tudatában az a kép alakult ki, hogy az említett korszerű programcsomagok használata már kellő biztonságot jelent bármilyen felhasználó számára. Többnyire magánbeszélgetések és saját tapasztalatok tanúsága szerint ez jelenleg még korántsem igaz, és igenis jónéhány „izzadságos" feladatmegoldás történik napjaink-

ban azért, mert a paraméterek választása nem megfelelő egy-egy feladatra, illetve azért, mert a jelenlegi LP eljárások nem képesek maguk ellátni a helyes választás feladatát.

Komoly segítséget jelentene, ha lenne egy olyan algoritmus, mely az aktuális feladatot analizálná és megfelelő kritériumok alapján optimálisan választaná meg a szimplex eljárás paramétereit, teljesen tehermentesítve azáltal a felhasználót és egyúttal megbízható eszközt jelentve bárki kezében. Igen előnyös lenne, ha a szimplex módszeren belül sikerülne találni olyan, hatékony algoritmikus technikákat, melyek nem, vagy alig érzékenyek a rendszerparaméterek változásaira.

A problémakör vizsgálatának fontosságára több tényező hívta fel a szerző figyelmét. 1969—70-ben az *Országos Tervhivatal* elkészítette a magyar népgazdaság IV. ötéves tervének lineáris programozáson alapuló közgazdasági modelljét. Az akkor elérhető számítógépek gyári LP programcsomagjai a hazai gépeken nem tudtak megoldani akkora méretű LP feladatokat, mint amekkorák az OT által felírt modellek voltak. (A megoldandó feladatok kb. 1000 kollektív feltételt, kb. 1800 változót és közel ezer egyedi felső korlátot tartalmaztak.) Az OT ezért megbízást adott olyan LP rendszer kidolgozására, mely képes megoldani a feladatot. A rendszer a *Nehézipari Minisztérium Ipargazdasági és Üzemszervezési Intézetében* készült el, és annak az optimalizáló részét a szerző készítette. A kezdetben nem sok sikerrel kecsegtető munka (kis kapacitású és nem túl gyors számítógép használata) végül eredménnyel zárult: a népgazdasági feladat 16 változatát sikerült megoldani. A feladatok — különösen akkoriban — tipikusan a nagyméretű LP feladatok területére estek. Tekintettel arra, hogy a szimplex módszer számítógépes realizációja (implementációja) területén abban az időben lényegében véve egyetlen értékelhető publikáció volt fellelhető (ORCHARD—HAYS [27]) és ez is elsősorban csak a főbb algoritmikus elemeket tárgyalta, jelentős fejlesztő munkára volt szükség. Ehhez kétségtelenül nagy segítséget nyújtott a népgazdasági modellel végzett számítógépes futtatások során szerzett rengeteg — negatív és pozitív — tapasztalat. Jellemző, hogy míg a modellnek 16 változata futott le, addig az optimalizáló rendszernek 25 változata készült el. Ez utóbbi nem is tekintendő különösen nagy számnak, hiszen nagyobb LP rendszerek száznál is több új kiadást érnek meg [7]. A munkával kapcsolatos eredményekről több publikáció jelent meg [13], [18], [6], [17].

Hatékony LP rendszerek elméleti és gyakorlati kérdéseinek a vizsgálata újabb esemény kapcsán kapott különös jelentőséget. Az ESZR program keretében sok új számítógépet helyeznek üzembe. Ezek a gépek egy számítógépcsalád, az R sorozat tagjai. A sorozat legkisebb gépei (R—10, R—12, újabban R—11 és R—15 is) Magyarországon készülnek. Ezek — különböző paramétereiknél fogva — igen alkalmasak vállalati információs rendszerek korszerű kezelésére és a műszaki-gazdasági tervező munka támogatására. Ez utóbbinak viszont fontos segédeszköze a lineáris programozás. Tekintettel arra, hogy a gépek alkalmazási programcsomagjai közül hiányzott az LP csomag, ezért a *KSH Országos Számítástechnika-alkalmazási Iroda* megbízást adott a *Számítógépalkalmazási Kutató Intézetnek* a programcsomag kidolgozására. A munka során lehetőség volt korábbi eredmények és tapasztalatok felhasználására, de számos új algoritmikus elem, eljárás kidolgozására és beépítésére is sor került. Az elkészített programcsomag a LIPROS (LI*near* PRO*gramming* S*ystem*) nevet kapta. A LIPROS-nál külön nehézséget jelentett az R—10 gépek kisméretű memóriája, amelyet még meg kellett osztani a program és az adattömbök

közt. Szerző az egész munka felelőse és egyben az optimalizáló rendszer elkészítője is volt. Az eredményekről számos előadás hangzott el és több publikáció jelent meg [19], [23], [24]. A LIPROS az 1980. évi akadémiai pályázat számítástechnikai kutatási főirányán első díjat nyert.

Az ESZR sorozat nagyobb gépei (R—20-tól fölfelé) alapvetően két operációs-rendszert használnak: a DOS, illetve az OS rendszert. A DOS-hoz a külföldi gyártó cég adott át egy LP2 jelű LP programcsomagot (ami egyébként az IBM által készített LPS-sel mutat rokon vonásokat), ez azonban rugalmas adatkezelő rendszere mellett számottevő lineáris programozási algoritmikus fogyatékosságokkal, sőt hibákkal terhesnek bizonyult, amint azt a honosítási munka során a szerzőnek sikerült kimutatni ([20] 5. fejezet). A probléma feltárása még nem jelentette azt, hogy a hibákat ki lehet javítani, ugyanis az LP2-re nincs a gyártó cég által támogatott programkövetés. Miután előreláthatólag a DOS rendszer még jó ideig számos gépen üzemelni fog, szükséges volt az LP2-nél jobb LP rendszer beszerzése illetve kidolgoztatása. Az R—10-re készült LIPROS-sal szerzett kedvező tapasztalatok eredményeként a LIPROS-on alapuló, de — kihasználva a nagyobb gépek lehetőségeit — annál robusztusabb változat elkészítése látszott olyan megoldásnak, mely mentessé tehető az említett hibáktól és ugyanakkor viszonylag rövid idő alatt használatba vehető. Ennek a munkának a kapcsán több érdekes eredmény született, melyek közül a legjelentősebb egy számos jó tulajdonsággal rendelkező, új első fázis eljárás ([25], [26] és jelen dolgozat 5.1—5.2 pontja) kidolgozása volt.

A DOS-ban működő új LP rendszer jelentőségét fokozza, hogy — jelen sorok írásakor — nincs még megnyugtatóan megoldva az OS-es LP rendszer kérdése. Amennyiben nem sikerül a gyártó cég által rendszeresen követett és karbantartott LP rendszert beszerezni, úgy az említett DOS-os LP rendszernek elkészíthető az OS-es változata, és ez használható egy IBM MPSX szintű nagy rendszer üzembe állításáig.

A saját készítésű rendszereknek, így egy ilyen LP rendszernek is, van egy nem elhanyagolható előnye: a rendszer teljes egészében kézben tartható. Ezen azt kell érteni, hogy a rendeltetéstől eltérő felhasználási szándék (pl. LP használata nemlineáris programozás során, LP dekompozíció kidolgozása) esetén lehetőség van a rendszerhez úgy hozzányúlni, hogy a kívánt felhasználás lehetővé váljék. Hasonlóképpen „saját" rendszer esetén lehetőség van az LP-t érintő lényeges, új elgondolások azonnali implementációjára is, amire más esetben nincs mód.

A lineáris programozás gyakorlati felhasználásának a jelentősége világszerte növekszik. Noha a jelenlegi LP rendszerek bizonyos igényeket rutinszerűen elég jól ki tudnak elégíteni, a kutató-fejlesztő munka egyáltalán nem szünetel. Ahhoz, hogy az LP programcsomagok ne csak a szakértők kezében legyenek jól használható eszközök még jelentősen növelni kell az LP rendszerek

— robusztusságát (megbízhatóságát, pontosságát),
— hatékonyságát,
— használati kényelmét.

(Ezen fogalmak pontos értelmezése a dolgozat 3. fejezetében található.)

Szerző abban a szerencsés helyzetben volt, hogy a fentiekre irányuló kutató-fejlesztő munkával jórészt hivatali feladatként, alkotó szellemű munkahelyi légkörben tudott foglalkozni. Tevékenységét viszont nehezítette és lassította az, hogy a

munkához nélkülözhetetlenül szükséges számítógéphasználat körülményei gyakran mostohák voltak (jelenlétes gépidők, bizonytalankodó számítógép, interaktív programfejlesztési lehetőség hiánya, stb.). Ezzel együtt a kutató-fejlesztő munka során az a kép alakult ki, hogy az LP rendszerekkel szemben támasztott fokozott követelmények kielégítésében jelentős szerep juthat az adaptivitás elve alkalmazásának, illetve olyan algoritmikus technikák kidolgozásának, melyek adaptív tulajdonságúak.

Ennek a témakörnek a tárgyalása óhatatlanul azt eredményezi, hogy bizonyos helyeken szólni kell az LP rendszerek implementálási részleteiről, illetve azokról a szempontokról, amelyeket az implementálás során figyelembe kell venni. Az ilyen részleteket a nagy LP rendszerek készítői bizonyos mértékig „műhelytitok"-nak tartják. Ez pedig azért van így, mert a lineáris programozásra is igaz az, hogy egy gyengébb algoritmus (vagy algoritmikus elemekből álló rendszer) jó implementációja sokkal jobb működési jellemzőkkel rendelkezhet, mint egy elvileg határozottan jobb algoritmus gyenge implementációja. Márpedig az alkotás értékelése szempontjából a működési jellemzőknek igen fontos a szerepe, hiszen a felhasználók elsősorban jól működő rendszerekkel akarnak dolgozni.

Az értekezés célja az, hogy feltárja az adaptivitás szerepét az LP rendszerekkel szemben támasztott követelmények fokozott kielégítésében. Ehhez szükség van a követelmények tételes kimunkálására, ami szintén a dolgozat feladata. Miután az adaptivitás elvének alkalmazása az LP implementálási fázisában kap fontos szerepet, ezért az implementációk főbb problémáiról is szólni kell. Az elmondottak konklúziójaként az adódik, hogy új LP algoritmikus elemek kidolgozásánál igen előnyös az, ha ezek minél több adaptív vonással rendelkeznek. A dolgozat utolsó fejezete éppen ilyen új eljárásokat mutat be.

Fentieknek megfelelően a dolgozat tagozódása a következő. A 2. fejezetben szerepel annak az LP feladatnak a felírása, melyre a későbbi tárgyalásban hivatkozás történik. Ugyancsak itt lesz szó a nagyméretű LP feladatok speciális tulajdonságairól és a módosított szimplex módszer számítógépes implementációjáról. Erre a fejezetre elsősorban a további tárgyalás előkészítése miatt volt szükség. A 3. fejezet az LP rendszerekkel szemben támasztott követelményekről és az adaptivitás elvének LP-beli alkalmazásáról szól. Itt szerepel annak a definíciója is, hogy jelen szövegösszefüggésben mit értünk adaptivitáson. A 4. fejezet az LP implementációk főbb problémáit vizsgálja. Az 5. fejezet a LIPROS-hoz készített három olyan algoritmikus elemet tárgyal, ahol az adaptivitás figyelembevételével jelentős javítást sikerült elérni a hasonló funkciókat ellátó eddigi elemekkel szemben. Ezek közül a legfontosabbnak az első fázisban a kilépő változó új meghatározási szabályát (DELPHI), valamint ugyanitt a belépő változó kiválasztására szolgáló új adaptív eljárást (ADOMCOMP) lehet tekinteni.

2. A szimplex módszer számítógépes implementációjáról

Ez a fejezet a szimplex módszernek azokat a főbb vonásait tárgyalja csak, amelyek a számítógépes implementáció szempontjából lényegesek. Új eredményeket természetesen nem tartalmaz, szemléletmódja azonban az implementációnál felmerülő problémák jellegének és súlyának jobb megértését kívánja elősegíteni.

2.1 *A megoldandó feladat alakja*

A kitűzött téma tárgyalásához először is szükség van annak az LP feladatnak a felírására, melyen vizsgálatainkat fogjuk végezni. Ez pontosan az az alak lesz, amellyel a számítógépes algoritmus is dolgozik és nem tartalmaz olyan egyszerűsítést, amelynek a megvalósítása a gyakorlatban valamilyen ok miatt nem célszerű, vagy értelmetlen (pl. egyedi felső korlátok 1-re normálása [27]).

Könnyen belátható, hogy a gyakorlatban előforduló legáltalánosabb LP feladatok az alábbi alakra hozhatók (pl. [27], vagy [16]):

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

és a változók az alábbi kategóriák valamelyikébe esnek:

típus	értéktartomány	
0	$y_i = 0$	$x_j = 0$
1	$0 \leqq y_i \leqq v_i$	$0 \leqq x_j \leqq u_j$
2	$0 \leqq y_i \leqq +\infty$	$0 \leqq x_j \leqq +\infty$
3	$-\infty \leqq y_i \leqq +\infty$	$-\infty \leqq x_j \leqq +\infty$

(2.2)

ahol u_j és v_i véges értékeket jelölnek. A (2.1) alatti feltételrendszer bármelyik sorát lehet célfüggvénynek tekinteni. A továbbiakban feltesszük, hogy a célfüggvény itt már úgy van felírva, hogy a hozzá tartozó y_i változót maximalizálni kell.

Számítógépes megoldás esetén a (2.1)—(2.2) alakra hozáshoz szükséges transzformációk az input során könnyen elvégezhetők és tárolhatók azok az információk, melyek a végeredmény visszatranszformálásához szükségesek.

A (2.1)-ben szereplő x_j változókat *strukturális változók*nak, míg az y_i-ket *logikai változók*nak nevezzük.

A strukturális változók típusainak értelmezése magától értetődő. A feltételekhez tartozó logikai változók típusa a feltételek típusától függ. Az egyenlőség típusú feltételekhez 0 típusú logikai változó tartozik, az eredetileg $\geqq$, vagy $\leqq$ típusú feltételekhez 2-es típusú logikai változó, míg a nem korlátozó típusú feltételek logikai változója 3-as típusú. Az ún. *„range"* típusú feltételek (amik valójában két feltételt jelentenek) logikai változója 1-es típusú és ez a következőt jelenti. Ha az eredeti feltétel

$$a_i \leqq \sum_{j=1}^{n} a_{ij} x_j \leqq b_i \tag{2.3}$$

alakú volt, akkor a logikai változó bevezetésével az

$$y_i + \sum_{j=1}^{n} a_{ij} x_j = b_i \tag{2.4}$$

és

$$0 \leqq y_i \leqq b_i - a_i$$

feltételek adódnak. Vagyis egy range típusú feltétel logikai változója éppen a feltételi intervallum határok közti távolságot futja be. A (2.4) átalakításnak az a lényege, hogy a (2.3)-ban felírt két — strukturális változókat tartalmazó — feltétel helyett egy kollektív és egy egyedi feltétel adódott. Tekintettel arra, hogy a tárgyalásban az egyedi felső korlát technikás szimplex módszer (pl. [16]) használatát tételezzük fel, így a (2.4) átalakítással egy bázisnövelő feltételt lehet megtakarítani.

A későbbiekben többször előnyös lesz a (2.1)—(2.2)-vel meghatározott feladatot olyan alakban felírni, ahol a logikai változókat 1-től m-ig indexezzük, míg a strukturálisakat $m+1$-től $m+n$-ig. Ugyanekkor az így adódó $N=m+n$ változót x_j-vel ($j=1, \ldots, N$) jelöljük és a véges egyedi felső korlátokat u_j-vel, függetlenül attól, hogy az előző felírás szerint logikai vagy strukturális változóról van szó. Az egységmátrixszal kibővített mátrixot **A**-val jelöljük.

Így az alábbi egységes alakot kapjuk:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{2.5}$$

ahol **A** mérete $m \times N$, **x** mérete N, míg **b** mérete változatlanul m és a változók az alábbi típusok valamelyikébe esnek:

típus	értéktartomány
0	$x_j = 0$
1	$0 \leqq x_j \leqq u_j$
2	$0 \leqq x_j \leqq +\infty$
3	$-\infty \leqq x_j \leqq +\infty$

(2.6)

A (2.5) feltételeket kollektív feltételeknek, a (2.6) feltételeket pedig egyedi feltételeknek nevezzük. A tárgyalásban mindig hivatkozni fogunk arra, hogy a (2.1)—(2.2) vagy a (2.5)—(2.6) alatti alakról van szó, így az azonos jelölések használata nem okoz kétértelműséget, ugyanakkor a képletek nagymértékben egyszerűsödnek ezáltal.

(2.5) egy bázisát jelöljük **B**-vel, a hozzátartozó bázismegoldást pedig $\boldsymbol{\beta}$-val:

$$\mathbf{B}\boldsymbol{\beta} = \bar{\mathbf{b}}, \tag{2.6a}$$

illetve

$$\boldsymbol{\beta} = \mathbf{B}^{-1}\bar{\mathbf{b}}. \tag{2.6b}$$

Itt $\bar{\mathbf{b}}$-sal jelöltük azt a jobboldalt, amelynél már figyelembe van véve a bázison kívül felső korláton levő változók hatása:

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

ahol J a bázison kívüli, felső korláton levő változók indexhalmaza, $\mathbf{a}_j$ pedig (2.5) j-edik oszlopa.

2.2 *Nagyméretű LP feladatok speciális tulajdonságai*

Elöljáróban le kell szögezni, hogy a „nagyméretű LP feladat" nem egy egzakt fogalom. Ez az elnevezés a számítástechnikai eszközökkel és az LP megoldó algoritmusokkal együtt jelentős tartalmi változáson ment át. Ami ugyanis nagyméretűnek minősült 1960 körül az akkori számítógépek, LP megoldó algoritmusok és a rajtuk nyugvó LP programcsomagok mellett, az pl. 1970 körül már legfeljebb csak közepes méretűnek volt tekinthető az új körülmények közt.

Pusztán a nagyságrendek érzékeltetésének kedvéért megemlítjük, hogy a mai — elsősorban hazai — gyakorlat mellett az 1000 kollektív feltétel körül van az a határ, amely fölött az LP feladatokat nagyméretűnek szokták nevezni.

A nagyméretű — de gyakran már az alig néhány száz feltételes — LP feladatoknak van egy olyan speciális tulajdonsága, amely a hatékony megoldás érdekében sok szempontból előnyösen kihasználható. Ez a tulajdonság pedig az, hogy az ilyen LP feladatok feltételi mátrixában a nullától különböző elemek száma az össz elemszámhoz képest kicsi, vagyis az ilyen mátrixok kis kitöltöttségűek, *ritkásak,* sűrűségük kicsi. A ritkásságnak sincs olyan értelemben vett egzakt definíciója, hogy hány százalékos kitöltöttség alatt nevezünk egy mátrixot ritkásnak. Jellemző értékek azonban ismeretesek. 1000 körüli feltételszám esetén a kitöltöttség általában 0,5% és 1% között mozog. Érdekes módon van egy olyan tapasztalat is [2], hogy élő feladatok feltételi mátrixai egy bizonyos méreten felül már általában olyanok, hogy oszlopként átlagban 5—10 nullától különböző elemet tartalmaznak. KALAN még azt is észrevette [15], hogy az amúgy is kevés nem-nulla elem között sok azonos értékű van. Ezt a tulajdonságot *szuper ritkás*ságnak *(super sparsity)* nevezzük.

A számítógépeken szokásos számábrázolások közül az egész típusú ábrázolás általában — néha lényegesen — kisebb helyen történik, mint a valós (lebegőpontos) típusú, így egy adott méretű memória területen több egész számot lehet tárolni, mint valóst. Ha tudjuk azt, hogy egész számaink nem haladnak meg bizonyos értéket (általában valamilyen alacsony 2 hatványt), az arány tovább javítható tömörített számábrázolás bevezetésével.

A most elmondottak azért érdekesek, mert a ritkás mátrixok tárolásánál csak a nem-nulla elemek (valós számok) és azok helymegjelölő adatai (egész számok) kerülnek be a gépbe. Ez azt jelenti, hogy például egy 1%-os sűrűségű 1000×1000-es feladat 1 millió adata helyett elég 10 000 valós és ugyanennyi egész számot tárolni indulásnál. A szuper ritkásság kihasználása esetén pedig, ha például 1000 különböző értékű mátrix bejegyzés van, ezt az 1000 valós értéket és kétszer 10 000 helymegjelölő egész számot (ez utóbbiakat esetleg tömörített formában) kell tárolni.

A szimplex módszernek a teljes tabló transzformációs változatánál a ritkásság viszonylag kevés iteráció alatt megszűnik, a tabló feltelik. Többek közt már ez a tulajdonság is alkalmatlanná teszi ezt a változatot nagyobb méretű feladatok megoldására, hiszen például egy 1 millió elemes tabló mozgatása iterációról iterációra a háttértároló és a központi egység közt rendkívül időigényes munka.

A szimplex módszer gépi realizációja lebegőpontos aritmetikát használ. Az ilyen aritmetikai műveletek kerekítési hibáktól és jegyveszteségtől terhesek. Különösen az additív típusú műveletek veszélyesek, mert ezek relatív hibája igen nagy lehet, ami pedig a szimplex eljárásra nézve komoly következménnyel járhat. Nagyméretű feladatoknál, ahol igen sok műveletet kell elvégezni a megoldás eléréséig, a veszélyek fokozott mértékben állnak fenn.

Összefoglalásképpen röviden megismételjük a nagyméretű LP feladatok itt tárgyalt tulajdonságait:

— a nagyméretű LP feladatok mátrixai ritkások, esetleg szuper ritkások,
— teljes tabló transzformációs szimplex eljárás alkalmazása esetén a tabló kezdeti ritkássága hamar megszűnik, a tabló feltelik,
— a nagyméretű LP feladatok hajlamosak a véges pontosságú numerikus számábrázolás és műveletvégzés miatt fellépő numerikus pontatlanságokra.

2.3 *A módosított szimplex módszer és számítógépes implementációja*

A (2.1)-ben felírt feltételrendszert tömör alakban így írhatjuk fel:

$$(\mathbf{I}, \mathbf{A})\begin{bmatrix}\mathbf{y}\\ \mathbf{x}\end{bmatrix} = \mathbf{b}, \tag{2.7}$$

ahol **I** az egységmátrix és a méretek (2.1)-nek megfelelően értendők. A (2.1)—(2.2) feladat megoldása a felsőkorláttechnikás szimplex módszer alapján végezhető el. Ekkor a munkabázis méretét csak a kollektív feltételek száma definiálja. Az egységmátrix jelenlétéből rögtön következik, hogy a feltételi mátrix rangja *m*. Jelöljük **B**-vel a (2.7) feltételrendszer egy bázisát. Az ehhez a bázishoz tartozó teljes transzformált tabló a

$$\mathbf{B}^{-1}(\mathbf{I}, \mathbf{A}) = (\mathbf{B}^{-1}, \tilde{\mathbf{A}}) \tag{2.8}$$

mátrixszorzással kapható meg a bázis inverzének ismeretében.

A módosított szimplex módszernek a lényege éppen az, hogy a szimplex iterációk végzéséhez szükséges információkat (a transzformált tabló megfelelő elemeit) az alapmátrix és a bázisinverz segítségével minden lépésben előállítja.

Így a nagyméretű LP feladatoknak az előző pontban említett első két tulajdonsága előnyösen használható ki, illetve vehető figyelembe, ugyanis így az alapmátrixra végig eredeti ritkás állapotában van szükség és miután a mátrix nem transzformálódik, az említett feltelítődés is elmarad. Ez tárolási és adatmozgatási szempontból feltétlenül előnyös, de az is könnyen belátható, hogy általában nem vezet számítási többletmunkához sem. Ez azonban még nem végleges megoldás, mert a mindenkori bázis (melynek rendje *m*) inverze éppen olyan mértékben feltelhet, mint a transzformált tabló. A számítógépes realizációk mégis szinte kivétel nélkül a módosított szimplex módszer variánsait használják. Ennek oka az, hogy az alapmátrix tárolása és mozgatása viszonylag jó hatásfokkal megoldható, ugyanakkor a bázis inverzének a gazdaságos kezelésére egy sor különleges eljárást dolgoztak ki, amelyekkel elérhető, hogy az inverzzel ekvivalens alakzat sűrűsége rossz esetben is alig haladja meg az alapmátrix sűrűségének néhányszorosát. Ez az inverzzel ekvivalens alak általában az inverz szorzat alakja (PFI, *Product Form of the Inverse*), vagy az inverz eliminációs alakja (EFI, *Elimination Form of the Inverse*). Valójában ez utóbbi kettős szorzat alak, mert a $\mathbf{B}=\mathbf{LU}$ felbontásban szereplő **L** alsó trianguláris, illetve **U** felső trianguláris mátrixok inverzét külön-külön szorzat formában célszerű tárolni. A PFI és EFI részletes tárgyalása az irodalomban megtalálható (pl. [27], [1]). A továbbiak bemutatásához elegendő, ha az egyszerű szorzat alakot (PFI) tekintjük át,

míg az EFI-nek csak néhány fontos tulajdonságát említjük meg. PFI esetén a $\mathbf{B}$ bázis $\mathbf{B}^{-1}$ inverzét $\mathbf{E}_i$ elemi transzformációs mátrixok (ETM) szorzataként írjuk fel:

$$\mathbf{B}^{-1} = \mathbf{E}_p \mathbf{E}_{p-1} \dots \mathbf{E}_1. \tag{2.9}$$

Itt p-vel a szorzat tényezőinek számát jelöltük. Az $\mathbf{E}_i$ mátrixok az egységmátrixtól egyetlen oszlopvektorban térnek el. Ezeket a vektorokat éta vektoroknak nevezzük.

$$\mathbf{E}_i = \begin{bmatrix} 1 & & \eta_1^{(i)} & & \\ & \ddots & \vdots & & \\ & & \eta_{r_i}^{(i)} & & \\ & & \vdots & \ddots & \\ & & \eta_m^{(i)} & & 1 \end{bmatrix}. \tag{2.10}$$

Az elemi transzformációs mátrixok tehát a sorszámukkal, az r_i indexszel és a $\boldsymbol{\eta}^{(i)}$ vektorral egyértelműen reprodukálhatók, így belőlük elég ezeket az információkat tárolni. Az r_i index azt mondja meg, hogy az $\boldsymbol{\eta}^{(i)}$ vektor $\mathbf{E}_i$-nek melyik oszlopában foglal helyet. A továbbiakban r_i-t a pivot sor indexének fogjuk nevezni. A $\mathbf{B}^{-1}$ képzésében szereplő éta vektorok összességét éta file-nak nevezzük.

A későbbiekben gyakran lesz szó az elemi transzformációs mátrixokkal végzett műveletekről. Ezek többnyire egy vektornak egy ETM-mel jobbról vagy balról való megszorzását jelentik. Az egyszerűség kedvéért jelöljük $\mathbf{E}$-vel az ETM-et, $\boldsymbol{\eta}$-val az éta vektort és r-rel a pivot pozíciót.

a) Oszlopvektor szorzása ETM-mel:

$$\begin{bmatrix} 1 & & \eta_1 & & \\ & \ddots & \vdots & & \\ & & \eta_r & & \\ & & \vdots & \ddots & \\ & & \eta_m & & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_r \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} a_1 + \eta_1 a_r \\ \vdots \\ \eta_r a_r \\ \vdots \\ a_m + \eta_m a_r \end{bmatrix}. \tag{2.10a}$$

Ha az eredményül adódó vektort $\hat{\mathbf{a}}$-val jelöljük, akkor érvényes az $\mathbf{E}\mathbf{a} = \hat{\mathbf{a}}$ összefüggés. A (2.10a) alakból azonnal adódik egy egyszerű, de nagyon fontos lemma:

2.3.1. LEMMA. Ha $a_r = 0$, akkor $\hat{\mathbf{a}} = \mathbf{a}$. Vagyis, ha a transzformálandó vektornak az ETM pivot pozíciójában levő eleme nulla, akkor a vektor változatlan marad, így egyetlen aritmetikai műveletet sem kell elvégezni rajta.

b) Sorvektor szorzása ETM-mel: $\hat{\mathbf{a}}^T = \mathbf{a}^T \mathbf{E}$

$$[a_1 \dots a_r \dots a_m] \begin{bmatrix} 1 & & \eta_1 & & \\ & \ddots & \vdots & & \\ & & \eta_r & & \\ & & \vdots & \ddots & \\ & & \eta_m & & 1 \end{bmatrix} = \begin{bmatrix} a_1 \\ \vdots \\ \sum_{i=1}^{m} a_i \eta_i \\ \vdots \\ a_m \end{bmatrix}. \tag{2.10b}$$

Ez az összefüggés azt mutatja, hogy $\hat{a}_i = a_i$, $i \neq r$ és $\hat{a}_r = \mathbf{a}^T\boldsymbol{\eta}$, tehát $\hat{\mathbf{a}}$ egyetlen „új" elemét egy skalárszorzással kapjuk.

A módosított szimplex módszer használata esetén minden egyes báziscsere alkalmával egy új elemi transzformációs mátrixszal bővül a (2.9) szorzat forma. Ily módon az inverzzel végzett műveletek is egyre több munkát igényelnek és egyre pontatlanabbak lesznek. Amikor mindez meghalad egy kritikus értéket, akkor célszerű végrehajtani az aktuális bázis újrainvertálását, vagyis a bázisvektorokból közvetlenül meghatározni a bázisinverz (2.9) szorzatalakját. Az újrainvertálás szerepéről, jelentőségéről és a vele szemben támasztott követelményekről a fejezet későbbi részében lesz szó.

A módosított szimplex módszer használata esetén szükség van a bázison kívüli vektorok árnyékárának meghatározására. A most következő részben a logikai és strukturális változók teljesen egyenrangú szerepet játszanak. Miután nem kívánunk köztük különbséget tenni, ezért az LP feladat (2.5)-beli alakját használjuk. Jelöljük I_B-vel a bázisváltozók indexhalmazát egy aktuális iteráció során, π-vel pedig a szimplex szorzót. A d_j $(j \notin I_B)$ árnyékárakat a nullától különböző típusú változókra a

$$d_j = \pi^T \mathbf{a}_j \tag{2.11}$$

skalárszorzással számíthatjuk ki, ahol $\mathbf{a}_j$ az LP feladat (2.5)-beli alakjában a j-edik oszlopot jelöli. Megfogalmazható olyan első fázisú célfüggvénytípus, hogy mind az első, mind a második fázisban azok lesznek a javítóvektorok az illetékes — első, vagy második fázisú — célfüggvény szerint, amelyekre

$$d_j < 0 \tag{2.12}$$

teljesül. Ha egy 1-es típusú változó a felső korlátján van, akkor ez a bázisba belépve negatív irányban mozdul el. Ahhoz, hogy ez a célfüggvény javulását eredményezze $d_j > 0$ szükséges. Hasonlóképpen, ha egy 3-as típusú változó negatív értékkel lép be a bázisba, $d_j > 0$ kell a javításhoz. Ezekben a jól definiált esetekben az árnyékárak -1-szeresét fogjuk jelölni d_j-vel és így a tárgyalás egyszerűsödik. Az így értelmezett d_j árnyékár rendszert korrigált árnyékár rendszernek is fogjuk hívni.

Az első fázisra vonatkozó állításnak a részletes vizsgálatát egy új típusú első fázisú eljárás bemutatásán keresztül az 5. fejezetben végezzük el. Második fázisban — maximalizálási feladatról lévén szó — (2.12) az ismert kritériumot jelenti. Ekkor ugyanis a szimplex szorzó a bázisinverznek annyiadik sorát jelenti, ahanyadik sor a célfüggvény a (2.5) alatti feltételrendszerben. Legyen ez a q-adik sor.

Ismeretes, hogy a bázisinverz q-adik sorát az

$$\mathbf{e}_q^T \mathbf{B}^{-1} \tag{2.13}$$

szorzással kaphatjuk, ahol $\mathbf{e}_q$ a q-adik m dimenziós egységvektor. Az első fázisbeli szimplex szorzó általában (az 5. fejezetben is) a bázisinverz sorainak valamilyen lineáris kombinációjából adódik. Ezt a lineáris kombinációt egy

$$\mathbf{e}^T \mathbf{B}^{-1} \tag{2.14}$$

típusú művelettel kaphatjuk meg, ahol $\mathbf{e}$ tartalmazza a megfelelő sorok súlyait.

(2.13)-at és (2.14)-et összegezve azt találjuk, hogy a π szimplex szorzó meghatározása a

$$\pi^T = \mathbf{v}^T \mathbf{B}^{-1} \tag{2.15}$$

összefüggéssel történik, ahol

$$\mathbf{v} = \begin{cases} \mathbf{c}, & \text{az első fázisban} \\ \mathbf{e}_q, & \text{a második fázisban.} \end{cases} \tag{2.16}$$

A d_j árnyékárak meghatározása után a következő lépésben meg kell vizsgálni, hogy volt-e olyan, nem 0 típusú változóhoz tartozó $j \notin I_B$ index, melyre $d_j < 0$ teljesült (természetesen a korrigált d_j-ről van szó). Ha nincs, akkor az aktuális célfüggvény szerinti optimumnál vagyunk. Ez a második fázisban egy optimális megoldás megtalálását jelenti, míg első fázisban vagy egy fizibilis megoldás megtalálását (ami után át lehet térni a második fázisra), vagy annak a detektálását, hogy nincs lehetséges megoldás. Ha van egy vagy több javító vektor, akkor közülük egyet kiválasztunk (pl. azt amelynek árnyékára a legnegatívabb) és képezzük a következő szorzatot:

$$\alpha_j = \mathbf{B}^{-1} \mathbf{a}_j, \tag{2.17}$$

amivel tulajdonképpen a (2.8) transzformált tabló j-edik oszlopát állítottuk elő.

Következő lépésként meg kell határozni a bázisból kilépő vektort, amit a továbbiakban pivot lépésnek nevezünk. Második fázis esetén itt derülhet ki, hogy a feladat megoldása nem korlátos. Felsőkorláttechnikás szimplex módszerről lévén szó, 1-es típusú változó esetén előfordulhat, hogy nincs kilépő vektor, vagyis nincs báziscsere, hanem a bevonni szándékozott változó átmegy egyik korlátjáról a másikra. A pivot lépés végrehajtása után kerül sor a megoldás transzformálására és a bázisinverz aktualizálására, amennyiben történt báziscsere. Mindezek után visszatérünk a szimplex szorzó meghatározásához és ismételjük a vázolt eljárás lépéseit.

Most nézzük meg, hogy a fenti lépések végrehajtása hogyan történik a szorzatalakú bázisinverz esetén.

A szimplex szorzót a (2.15) összefüggésből a (2.9) szorzatforma alapján határozzuk meg:

$$\pi^T = \mathbf{v}^T \mathbf{B}^{-1} = \mathbf{v}^T \mathbf{E}_p \ldots \mathbf{E}_1. \tag{2.18}$$

Ez a transzformáció a szakirodalomban BTRAN *(Backward* TRAN*sformation)* néven ismeretes. Az elnevezés arra utal, hogy az elemi transzformációs mátrixok éta vektorait a keletkezésükkel ellentétes sorrendben vesszük elő, ugyanis (2.18) valójában a

$$\pi_k^T = \pi_{k-1}^T \mathbf{E}_{p-k+1}, \quad k = 1, \ldots, p \tag{2.19}$$

rekurzió végrehajtását jelenti, ahol $\pi_0^T = \mathbf{v}^T$, és π_p^T éppen a keresett π^T lesz.

A szimplex szorzóval végzett (2.11) típusú műveletnél a bázisinverznek nincs szerepe, így ez a lépés változatlan marad.

A bázisba belépő $\mathbf{a}_j$ vektor betranszformálása (2.17) és (2.9) alapján az

$$\alpha_j = \mathbf{B}^{-1} \mathbf{a}_j = \mathbf{E}_p \ldots \mathbf{E}_1 \mathbf{a}_j \tag{2.20}$$

alakban írható fel. Ennek a műveletnek FTRAN *(Forward* TRAN*sformation)* a szokásos neve és arra utal, hogy az éta vektorokat természetes sorrendben kell elővenni.

A pivot lépésnél a bázisinverznek ismét nincs közvetlen szerepe, így a lépés végrehajtása változatlan marad.

Amennyiben volt báziscsere, a transzformációs lépésben először elkészül a (2.9) szerinti új elemi transzformációs mátrix (illetve éta vektor) $\mathbf{E}_{p+1}$, majd transzformálódik a megoldás, végül pedig a bázisinverz aktualizálása gyanánt $\mathbf{E}_{p+1}$ felkerül a szorzatalakú bázisinverz szorzótényezői közé. Ha $\hat{\mathbf{B}}$-val jelöljük az új bázist, akkor ennek inverze

$$\hat{\mathbf{B}}^{-1} = \mathbf{E}_{p+1}\mathbf{B}^{-1} = \mathbf{E}_{p+1}\mathbf{E}_p \ldots \mathbf{E}_1 \tag{2.21}$$

lesz. Ha nincs báziscsere, akkor a helyzet sokkal egyszerűbb, nem keletkezik új éta vektor, csak a megoldás egyszerű transzformációjára kerül sor.

A további hivatkozások egyszerűsítése érdekében most röviden összefoglaljuk a felsőkorlát technikával dolgozó szorzatformás módosított szimplex módszer egy iterációra eső főbb lépéseit.

1. lépés: Szimplex szorzóhoz szükséges $\mathbf{v}$ vektor elkészítése (2.15) és (2.16) alapján, az aktuális helyzetnek megfelelően (I. vagy II. fázis).
2. lépés: $\pi^T = \mathbf{v}^T\mathbf{B}^{-1}$ számítása BTRAN-nal.
3. lépés: A nem 0 típusú, bázison kívüli változókra a korrigált árnyékár meghatározása a $d_j = \pi^T\mathbf{a}_j$ összefüggés alapján, több javító vektor esetén a megfelelő kiválasztása. (A „megfelelő" értelmezését egyelőre nyitva hagyjuk.)
4. lépés: Ha nincs javítóvektor, akkor az eddig elért eredmény értékelése (megoldás fizibilis, vagy infizibilis) és megállás. Ha van javítóvektor, akkor következik az 5. lépés.
5. lépés: A kiválasztott $\mathbf{a}_j$ vektor betranszformálása az aktuális bázisba

$$\alpha_j = \mathbf{B}^{-1}\mathbf{a}_j$$

FTRAN-nal.

6. lépés: Pivot lépés, vagyis annak eldöntése, hogy a belépő vektor melyik bázisbelivel cserél helyet, illetve van-e egyáltalán báziscsere. (Itt derülhet ki a megoldás nem-korlátos volta is.)
7. lépés: Transzformációs lépés. Ha volt báziscsere, akkor új éta vektor képzése, az éta file kiegészítése, megoldás transzformációja. Ha nincs báziscsere, akkor megoldás transzformáció. Visszatérés az 1. lépésre.

Ezen 7 lépéssel jellemzett algoritmusra a továbbiakban RSM1 néven fogunk hivatkozni.

Miután általában egy RSM1 típusú algoritmus váz képezi a számítógépes implementációk alapját, érdemes egy kicsit elemezni a működését. Előljáróban meg kell még jegyezni, hogy a nagyméretű feladatok **A** feltételi mátrixát és a bázisinverzet reprezentáló éta file-t általában a számítógép nagykapacitású háttértárolóján (mágneslemez, esetleg dob) lehet csak tárolni. A háttértárolóval történő adatkommunikáció azonban legalább 2 nagyságrenddel lassúbb, mint a központi tárban levő adattömbökkel.

Az 1. lépéshez lényegében véve csak az aktuális bázismegoldásra és néhány ehhez kapcsolódó információra van szükség (bázisváltozók indexei, típusai, felső korlátai). A lépés végrehajtása logikailag egyszerű, adatmozgatási igénye igen csekély, vagy esetleg semmi (implementációtól függően).

A 2. lépés logikailag nem bonyolult, adatmozgatási igénye (az éta vektorokat keletkezésükkel ellentétes sorrendben kell behozni) azonban nagy és az aritmetikai műveletek száma is jelentős. A BTRAN-ban végzett skalárszorzás hajlamos a numerikus pontatlanságra, ezért precíz implementálást igényel. Az egész lépés elég lassú — sőt az éta vektorok növekvő számával egyre jobban lassul — és ugyanakkor pontatlanná is válhat.

A 3. lépésnél — noha csak a bázison kívüli vektorok jönnek számításba, azok közül is csak a nem 0 típusúak — általában az egész mátrix file inputjára sor kerül. A logikai vektorok fizikai tárolása természetesen nem történik meg, így azokkal kapcsolatban nincs adatmozgás. Mivel a feltételi mátrix a szimplex iterációk során változatlan formában marad, ezért ennek a lépésnek a munkaigénye állandó. A lépés logikai műveletigénye közepes, az aritmetikai műveletek száma a mátrix nem-nulla elemeinek számával arányos.

A 4. lépés az iterációk során nem munkaigényes.

Az 5. lépés, az FTRAN végrehajtása ismét a teljes éta file behozását igényli, tehát input igénye a báziscserék során állandóan növekszik. Ezzel együtt egyre több aritmetikai műveletet kell elvégezni, ami ugyancsak lassulást eredményez és a pontosság romlását okozhatja. Ugyanakkor ez a lépés logikailag nem bonyolult.

A 6. lépésnek alig, vagy egyáltalán nincs adatmozgatási igénye, az aritmetikai műveletek száma is csekély. Logikailag viszont ez a lépés a legbonyolultabb, amint erről a későbbiekben bőven lesz szó.

A 7. lépésben nincs számottevő adatmozgatás, sem különösebben sok aritmetikai művelet, és a műveletek logikája sem bonyolult. Az iterációt lezáró adminisztráció azonban viszonylag sokrétű, így ha végrehajtásban nem is, de gépi programban ez egy hosszú rész.

Az egy iterációra eső lépések vizsgálata végülis azt mutatja, hogy az iterációs időigény szempontjából alapvető szerepe van az éta file-nak. Az éta file-t ugyanis egymástól függetlenül kétszer kell „áthúzni" a memórián (ráadásul egyszer elölről, egyszer hátulról kezdve), ami hosszú éta file esetén nagy munkát jelent. Hasonlóképpen nagy hatással van az éta file a számítási pontosságra is, ugyanis az egyre később készülő éta vektorok hordozzák elődeik hibáit és általában még hozzá is adnak valamennyit, aminek következtében a bázisinverzzel végzett műveletek egyre pontatlanabbak lesznek.

A módosított szimplex módszer egyik igen előnyös tulajdonsága az, hogy bármikor lehetőség van a szimplex iterációk hibáitól terhes bázisinverz közvetlen, újbóli és várhatóan pontosabb meghatározására az aktuális bázis ismeretében. Ezt a lépést *újrainvertálás*nak nevezzük. A pontosság növekedésére az ad jó esélyt, hogy az újrainvertálás szabad paramétereit (pl. pivot sorok sorrendje, oszlopcsere) nagy mértékben a fokozott numerikus pontosság érdekében lehet felhasználni. Ugyanerre a szimplex iterációk során nincs lehetőség, mert ott a konkrétan alkalmazott LP pivot kritérium (pl. fizibilitás fenntartása) határozza meg az új éta vektort és ezen keresztül az új bázisinverzet.

A szorzatformás bázisinverz alkalmazása esetén újrainvertáláskor ki lehet

használni azt, hogy a szorzatalak nem egyértelmű, noha a tényezők száma adott, ugyanis a bázisbeli strukturális változók számával egyenlő. Megfelelő sorrend megválasztásával elérhető, hogy a keletkező éta file hossza (vagyis az éta vektorokban a nemnulla elemek száma) kicsi, esetleg minimális legyen. Ennek előnyei az RSM1 algoritmus szempontjából a következők:

— az éta vektorok elemei numerikusan pontosabbak lesznek,
— a rövidebb éta file miatt a BTRAN és az FTRAN műveletek az újrainvertálás után gyorsabbak lesznek,
— az éta vektorok kevesebb nemnulla eleme miatt a velük végzett műveletek során kevesebb aritmetikai műveletet kell elvégezni, ami az eredmény pontosságára van kedvező hatással.

Az természetesen az újrainvertálás után is igaz marad, hogy a soron következő szimplex iterációk által generált éta vektorok telítettsége nagy lehet, hiszen — mint ismeretes — egy éta vektor (illetve elemi transzformációs mátrix) az α transzformált oszlopból az alábbiak szerint képződik:

$$\mathbf{E} = \begin{bmatrix} 1 & & \eta_1 & & \\ & \ddots & \vdots & & \\ & & \eta_r & & \\ & & \vdots & \ddots & \\ & & \eta_m & & 1 \end{bmatrix}, \tag{2.22}$$

ahol

$$\eta_i = -\alpha_i/\alpha_r, \quad i \neq r \tag{2.23}$$

$$\eta_r = 1/\alpha_r$$

és r a pivot sor indexe.

Ebből látható, hogy az éta vektor kitöltöttsége éppen az α vektor kitöltöttségével lesz azonos.

Az éta file gyors növekedése miatt gyakran van szükség újrainvertálásra, az éta file rövidebbé és pontosabbá tételére. Az újrainvertálásnak ily módon jelentős szerepe van a szorzatformás módosított szimplex módszernél. A vele szemben támasztott főbb követelmények:

— rövid éta file generálása,
— pontos éta file generálása,
— gyors működés.

Ez utóbbi azért lényeges, mert újrainvertáláskor az optimalizálás szempontjából nem történik előrehaladás, mindössze az aktuális állapot reprodukciója és ha az újrainvertálások időigénye nagy lenne, az nagyon hátráltatná az optimalizálás menetét. Éppen ezért nagy erőfeszítések történtek a fenti követelményeknek minél jobban eleget tevő invertáló eljárások kifejlesztése érdekében. Ennek eredményeképpen ma már igen hatékony szorzatformás invertáló algoritmusok léteznek ([27], [11], [12]), melyek igen kis kitöltöttségű, a bázis kis kitöltöttségének alig néhányszorosát elérő sűrűségű éta file-t tudnak gyorsan és pontosan előállítani. Példának megemlítünk

néhány tipikus értéket: 0,2%—1,0%-os sűrűségű bázishoz általában 0,5%—4,0%-os éta file generálódik.

Ezen a helyen célszerű felidézni azt, hogy a bázisban levő egységvektorokból újrainvertáláskor nem keletkezik éta vektor, tehát a szorzatalak tényezőinek a száma a bázisbeli strukturális oszlopok számával lesz egyenlő. (Ennek igazolása pl. [9]-ben megtalálható.) Tekintettel arra, hogy valós feladatok bázisaiban a logikai változók aránya 10—35% között szokott mozogni, ezért újrainvertálás után az éta vektorok száma (p) akár lényegesen is kevesebb lehet mint a bázis mérete (m).

A bázisinverz viszonylag újabban kifejlesztett eliminációs alakja a bázis $\mathbf{B}=\mathbf{LU}$ alakú trianguláris felbontásából indul ki, ahol $\mathbf{L}$ alsó, $\mathbf{U}$ pedig felső trianguláris. Ezután mind az $\mathbf{L}$, mind az $\mathbf{U}$ tényezőt szorzat alakban írja fel, ami által egy kettős szorzat alak adódik. Ennek a módszernek az egyik előnye akkor mutatkozik meg, amikor a bázis kombinatorikusan csak kevéssé triangularizálható, ilyenkor ugyanis a normál szorzatalaknál rövidebb és általában pontosabb éta file készíthető. Másik, nem kevésbé fontos előnye pedig az, hogy lehetséges a triangularitás fenntartása a szimplex iterációk során. Ezzel egyidejűleg az éta file növekedése sokkal lassúbb, mint PFI-nél, így több iterációt lehet végezni újrainvertálás nélkül. Mindezekért az előnyökért az EFI és a soron következő szimplex iterációk bonyolultabb logikájával és esetleg kicsit lassúbb működésével kell fizetni.

Az invertálásról és az éta file szerepéről eddig elmondottak elegendőek ahhoz, hogy RSM1-et újraértékeljük és bevezessünk egy fontos algoritmikus elemet.

RSM1 2. lépésében a teljes éta file-t be kell egyszer (de nem egyszerre, hanem általában részekre tördelten) hozni a központi memóriába. A 3. lépésben a teljes feltételi mátrixot kell behozni a háttértárolóról, és a mátrix átnézésének eredményeképpen kiválasztunk egyetlen javítóvektort az esetleg nagyon sok potenciális javítóvektor közül. Az 5. lépésben pedig ismét az egész éta file behozására van szükség az egy javítóvektor betranszformálására.

A *többszörös kiválasztás (multiple pricing)* [27] elvének alkalmazása esetén pontosan ugyanennyi háttérkommunikációval egyszerre több javítóvektort választunk ki és állítjuk elő FTRAN-nal azoknak az aktuális bázisbeli képét. Így az egy betranszformált vektorra eső munka csökken. Az egy iterációra eső tényleges munka csökkenését számos további tényező jelentősen befolyásolja. Jelöljük K-val a kiválasztásra engedélyezett vektorok maximális számát, k-val pedig a tényleges kiválasztottakét, nyilván $k \leqq K$. Multiple pricing esetén a szimplex eljárás úgy folytatódik, hogy a $j_1, \ldots, j_k$ indexű kiválasztott változók és az aktuális bázisváltozók által meghatározott redukált LP feladaton dolgozunk. Szokás ezt az eljárást szuboptimalizálásnak is nevezni. (A redukált LP feladat meghatározásában valójában még a nemkiválasztott és nem-bázisbeli változók is szerepet játszanak, nevezetesen azok, melyek 1-es típusúak és az aktuális állapotban a felső korlátjukon vannak. A szuboptimalizálás alatt ezek állapota azonban nem változik.) A szuboptimalizálásban résztvevő vektorok körében már van lehetőség a legmeredekebb irány helyett a legnagyobb előrehaladás elvének az érvényesítésére, most ugyanis az ehhez szükséges információkat — a betranszformáláshoz képest — csekély többletmunkával elő lehet állítani. Az $\alpha_{j_1}, \alpha_{j_2}, \ldots, \alpha_{j_k}$ vektorok valójában a redukált LP feladat transzformált tablóját jelentik. Szuboptimalizálás során közülük egyet bevonunk az aktuális bázisba, ami által keletkezik egy újabb éta vektor, amit elhelyezünk az éta file-on. (Azt az egyszerű esetet most nem tárgyaljuk, amikor csak korlátcsere történik, bázis-

csere nélkül.) Könnyen belátható, hogy a bázisból kilépő vektor új bázisbeli ($\hat{\mathbf{B}}$-beli) képe éppen az új éta vektorral ($\boldsymbol{\eta}^{(p+1)}$) azonos. Ha ugyanis egy $\mathbf{a}_j$ vektor az r-edik bázisváltozóval (x_{i_r}) cserél helyet, mely utóbbihoz tartozó oszlopvektor $\mathbf{a}_{i_r}$, akkor

$$\hat{\mathbf{B}}^{-1}\mathbf{a}_{i_r} = \mathbf{E}_{p+1}\mathbf{B}^{-1}\mathbf{a}_{i_r} = \mathbf{E}_{p+1}\mathbf{e}_r = \boldsymbol{\eta}^{(p+1)}. \tag{2.24}$$

Az egyenlőség első része (2.21) miatt igaz, a második rész abból következik, hogy $\mathbf{a}_{i_r}$ a $\mathbf{B}$ bázis r-edik eleme így $\mathbf{B}$-beli képe éppen $\mathbf{e}_r$, míg az utolsó egyenlőség (2.10) értelemszerű alkalmazásával adódik, figyelembe véve, hogy $\mathbf{E}_{p+1}$ pivotpozíciója éppen r.

A szuboptimalizálásban résztvevő vektorokat — az éppen kiválasztott kivételével — és a megoldást transzformáljuk $\mathbf{E}_{p+1}$-gyel, ami valójában egy egyetlen éta vektoros FTRAN-t jelent. Ezután újra kiértékeljük a helyzetet: először a megoldás fizibilitását vizsgáljuk meg (infizibilis: első fázis; fizibilis: második fázis) és utána az aktuális állapotnak megfelelő célfüggvény szerint meghatározzuk a szuboptimalizálásban résztvevő változók árnyékárait. Ha csak egy javítóvektor is akad köztük, akkor ismét egy szuboptimalizáláson belüli ún. *minor iteráció* következik az előbbiek szerint. Ha egy javítóvektor sem adódik, akkor visszatérünk az RSM1 szerinti 1. lépésre és elkezdünk egy többszörös kiválasztás szerinti fő, más néven *major iterációt*.

RSM1-nek a többszörös kiválasztással és az újrainvertálással kibővített változatát, amelyet RSM2-vel jelölünk, az alábbiakban foglaljuk össze. Egyben, hogy a lépések közötti logikai kapcsolatot ábrán is szemléltetni tudjuk, minden lépésnek adunk egy emlékeztető elnevezést.

1. lépés: (FEASIBILITY): Szimplex szorzóhoz szükséges $\mathbf{v}$ vektor előállítása (mint RSM1/1 lépés).
2. lépés: (BTRAN): $\boldsymbol{\pi}^T = \mathbf{v}^T\mathbf{B}^{-1}$ számítása BTRAN-nal (mint RSM1/2. lépés).
3. lépés: (MPRICE, a multiple pricing rövidítéseként): A nem 0 típusú, bázison kívüli változókra a korrigált árnyékár meghatározása $d_j = \boldsymbol{\pi}^T\mathbf{a}_j$ alapján és a $k \leqq K$ — valamilyen további szempont szerint — legkedvezőbb $\mathbf{a}_{j_1}, \ldots, \mathbf{a}_{j_k}$ vektor kiválasztása.
4. lépés: (CHECK) Megnézzük, hogy volt-e egyáltalán javítóvektor, vagyis $k > 0$? Ha igen, akkor rátérünk az 5. lépésre. Ha nem, akkor következik az eddig elért eredmény értékelése (megoldás fizibilis, vagy infizibilis) és megállás.
5. lépés: (MFTRAN, a multiple FTRAN rövidítéseként): A kiválasztott $\mathbf{a}_{j_1}, \ldots, \mathbf{a}_{j_k}$ vektorok betranszformálása az aktuális bázisba:

$$\begin{aligned} \boldsymbol{\alpha}_{j_1} &= \mathbf{B}^{-1}\mathbf{a}_{j_1} \\ &\vdots \\ \boldsymbol{\alpha}_{j_k} &= \mathbf{B}^{-1}\mathbf{a}_{j_k} \end{aligned}$$

az oszlopokon párhuzamosan végrehajtva az FTRAN műveletet.

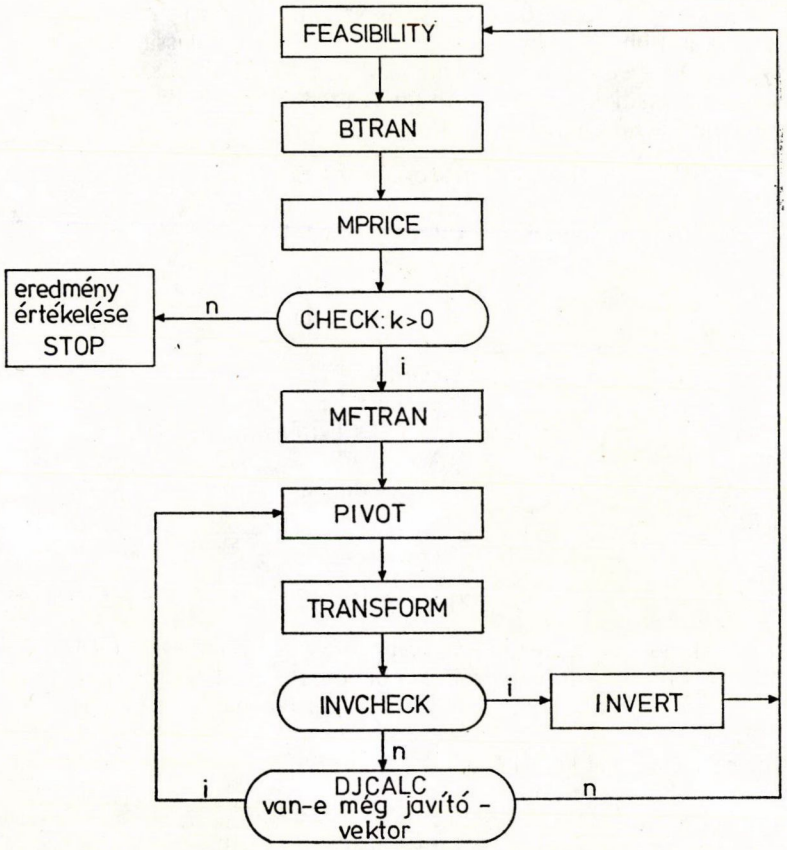
6. lépés: (PIVOT): A javítóvektorok közül annak kiválasztása (a hozzá tartozó kilépő vektor meghatározásával együtt), melynek a bázisba történő belépése az aktuális célfüggvény szerint a legnagyobb előrehaladást biztosítja; nem-korlátosság kiderítése.
7. lépés: Most több részre bomlik:
   7a. (TRANSFORM): Azonos az RSM1 szerinti 7. lépéssel, de most még a szuboptimalizálásban résztvevő vektorokat is transzformálni kell $\mathbf{E}_{p+1}$-gyel, ha egyáltalán keletkezett $\mathbf{E}_{p+1}$.

7b. (INVCHECK): Annak ellenőrzése, hogy kell-e újrainvertálni. (A kritérium kérdését egylőre nyitva hagyjuk.) Ha igen, akkor az invertálás végrehajtása (INVERT) és utána visszatérés az 1. lépésre. Ha nem, akkor:

7c. (DJCALC): A szuboptimalizálásban résztvevő és betranszformált vektorokra a transzformáció után aktuális célfüggvény szerint a d_j korrigált árnyékárak meghatározása. Ha van még javítóvektor, akkor visszatérés a 6. lépésre, különben az 1. lépésre.

A lépések közti logikai kapcsolatot a következő ábrán mutatjuk be.

Már RSM1-ben is volt olyan nyitott algoritmikus kérdés (a 3. lépésben a „megfelelő" javítóvektor kiválasztása a sok — matematikailag egyébként szintén jó — javítóvektor közül), amelynek konkrét megválaszolásától nagy mértékben függhet a megoldás eléréséhez szükséges munka. Az RSM2 jelű algoritmus-váz az említett megfontolások miatt jobbnak (de legalább is flexibilisebbnek) mondható, ugyanakkor még több hasonló jellegű algoritmikus kérdést hagy nyitva, mint RSM1,



1. ábra Az RSM2 algoritmus blokkdiagrammja

például: mennyi legyen a kiválasztható vektorok maximális K száma; van-e értelme, s ha igen, milyen módon, korlátozni a szuboptimalizálásban végzett minor iterációk számát; a szuboptimalizálásban a közel egyformán jó javítóvektorok közül ténylegesen melyik lépjen be a bázisba.

Ezzel tulajdonképpen csak az RSM2-ben található néhány nyitott algoritmikus kérdésre hívtuk fel a figyelmet, ugyanakkor még távolról sem érintettük a numerikus pontosságot érintő kérdéseket, az itt még nem is említett algoritmikus technikák által kínált további szabad választási lehetőségekben rejlő potenciális előnyöket — a buktatóikkal együtt.

Ahhoz, hogy a már feltett és még ki nem mondott kérdésekre választ tudjunk adni, szükséges pontosan felmérni és megvizsgálni azokat a szempontokat, amelyek tükrében az egész LP eljárás, vagy annak részei működését el akarjuk bírálni, illetve tisztázni azokat a követelményeket, amelyeket az LP rendszerek képességeivel szemben támasztunk. Így lehetőségünk lesz a paraméterválasztás jóságát mérni, vagy kitűzni azokat a célokat, amelyeket a kifejlesztendő új algoritmikus technikáknak el kell érniük.

A követelmények ilyen jellegű tételes ismerete azért is fontos, mert ily módon könnyebben el tudjuk dönteni egy LP rendszerről, hogy céljainknak megfelel, vagy sem.

3. LP rendszerekkel szemben támasztott követelmények és az adaptivitás

3.1. *LP rendszerekkel szemben támasztott követelmények*

Az LP rendszerek képességeit felhasználói szempontból elsődlegesen két paraméterrel szokták jellemezni, úgymint: 1. mi a ténylegesen megoldható feladatok maximális mérete, 2. mennyi a megoldás eléréséhez szükséges gépidő (vagy ami ezzel általában arányos: mekkora a megoldás költsége). Az igényesebb felhasználót esetleg még az is érdekli, hogy mennyire lesz pontos az eredmény.

Valójában ezen igények mögött mélyebb követelmények is rejlenek. Ugyanis ha például azt a kérdést tesszük fel, mennyibe fog kerülni egy adott feladat megoldása, akkor ebbe implicite beleértjük azt is, hogy az LP rendszer egyáltalán képes megoldani a feladatot, ami viszont nem egy magától értetődő dolog egy olyan numerikus eljárásnál, mint a szimplex módszer. Éppen ezért feltétlenül szükséges áttekinteni az LP rendszerekkel szemben támasztott legfontosabb követelményeket.

A most tárgyalandó, az LP rendszerek jellemzésére szolgáló fogalmak meghatározása nem tekinthető matematikai értelemben egzakt definíciónak. A fogalmak jelentősége abban áll, hogy segítségükkel össze lehet hasonlítani különböző LP rendszereket, LP technikákat, így lehetővé válik a felhasználó számára az orientálódás, ugyanakkor érthetőbbek lesznek azok az erőfeszítések — így jelen dolgozat 5. fejezete — is, amelyek az LP rendszerek jobbítására törekednek.

Fontos jellemzője ezeknek a fogalmaknak, hogy segítségükkel az LP rendszerek relatív értékelésére van lehetőség és az értékelés eredménye függ azon feladatoktól, melyek megoldására sor került az összehasonlító vizsgálatok elvégzésekor. Erre azért kell előre felhívni a figyelmet, hogy megfelelően lehessen értelmezni az adódó következtetések érvényességi körét.

Megjegyzendő, hogy numerikus eljárások értékelésére nem ismeretes olyan általánosan használható módszer, mellyel erősebb állításokat lehetne kimondani. Éppen ezért — akárcsak a numerikus kvadratúra területén — nagy jelentősége van egy helyesen megválasztott *tesztfeladattár*nak, amely alapját képezi a különböző LP rendszerek összehasonlításának, illetve lehetővé teszi, hogy egy LP rendszer algoritmikus fejlesztésének a hatását le lehessen mérni. A tesztfeladattár elkészítésének szempontjaival itt nem kívánunk foglalkozni, pusztán annyit említünk meg, hogy minél több feladattípus reprezentánsai szerepelnek a feladattárban, annál szélesebb körre vonatkozó következtetéseket lehet levonni, ugyanakkor viszont annál költségesebbek lesznek a vizsgálatok.

Az LP rendszerekkel szemben támasztott követelmények közt első helyen kell megemlíteni a *megbízhatóságot*.

A *megbízhatóság* a sikeres feladatmegoldások számának és az összes megoldási kísérletek számának arányát jelenti. Egy feladatmegoldást sikeresnek nevezünk, ha az eljárás egy olyan megoldást szolgáltatott, amely a matematikailag helyes eredménytől legfeljebb egy bizonyos tűrési hibával tér el. A megbízhatóságon belül alapkövetelmény az, hogy az LP rendszer minden feladat — akár tévesen megadott, értelmetlen feladat — esetén értelmes illetve értelmezhető választ adjon, ne kezdjen definiálatlan tevékenységbe (pl. rosszul megadott futási paraméterek esetén), ne essen numerikus vagy algoritmikus végtelen ciklusba. Bár ezek a követelmények eléggé triviálisnak tűnnek, teljesítésük egyáltalán nem magától értetődő és könnyű, amint azt néhány működő nagy LP rendszernél (pl. LP2 [20]]) tapasztalni lehet.

Egy LP rendszer *megbízhatóbb* egy másik LP rendszernél, ha a tesztfeladatok — esetleg azokon túli, de mindkettő által megoldott feladatok — közül többet tud sikeresen megoldani és várhatóan megbízhatóbban működik hasonló összetételű feladatok sokaságán.

A *pontosság* követelményét úgy értelmezzük, hogy az az eljárás a *pontosabb,* amelyik a megoldást több pontos jeggyel tudja szolgáltatni. (Ennek természetesen határt szab az adott számítógép aritmetikájának pontossága.)

Ha egy LP rendszer a megoldást egy másik rendszernél megbízhatóbban és pontosabban képes szolgáltatni, akkor azt mondjuk, hogy az első rendszer *robusztusabb* a másiknál. Minél robusztusabb tehát egy LP rendszer, a feladatoknak annál szélesebb skáláját képes megoldani megfelelő pontossággal.

A követelmények közt a következő helyen a hatékonyságot kell említeni. A *hatékonyság* fordítottan arányos az egy LP feladat megoldásához szükséges munka és/vagy erőforrás igénybevétel illetve ráfordítás nagyságával. Ilyen értelemben egy LP rendszert egy adott feladatra nézve *hatékonyabb*nak mondunk egy másiknál, ha kevesebb munkával és/vagy erőforrás felhasználásával tudja megoldani a feladatot.

A hatékonyság mérésére szolgáló munkát ki lehet fejezni a megoldás eléréséhez szükséges elemi (vagy összetett) műveletek számával, futási idővel (processzor [CPU] idő, adatátviteli [channel] idő), esetleg a megoldás költségével. Az összetett műveletek közül gyakran szokás a szimplex iterációk számát nézni (esetleg különválasztva a báziscseréket és a puszta korlátcseréket). Az iterációszám azonban csak megfelelő körültekintéssel használható különböző LP rendszerek összehasonlítására, ugyanis az egy iterációra eső munka nagyon eltérő lehet az alkalmazott megoldási technikáktól függően. Jelentősége azonban mindmáig megmaradt mégpedig elsősorban azért, mert értéke rendkívül egyszerűen adódik a végeredménnyel együtt, továbbá az alkal-

mazott technikák ismeretében többé-kevésbé el lehet dönteni, hogy az egy iterációra eső munka az összehasonlítandó rendszereknél lényegesen különböző, vagy közel egyenlőnek tekinthető és ugyanakkor lényegében független az alkalmazott számítógéptől.

Látszólag jobb mérőszámnak kínálkozik a CPU és channel idő, vagy ezek kombinációja. Sajnos azonban ez sem bizonyul egyértelmű mérőszámnak. A multi üzemmódban dolgozó számítógépek esetén ugyanis a gép aktuális leterheltségétől függően ugyanannak a feladatnak a megoldásához szükséges CPU idő szélsőséges esetben 20—30%-os eltérést is képes mutatni [14]. LP rendszerek önálló (mono) üzemmódban történő futtatására gyakorlatilag szinte semmi reális lehetőség nincs [14], így a külső körülményektől független, pontos CPU idők sorozatos mérésére a gyakorlatban nem lehet számítani. Éppen ezért nem is lenne reális célkitűzés olyan LP rendszert készíteni, mely mono üzemmódban működik csak kedvezően (pl. az összes gépi erőforrás egyidejű igénybevételével). A CPU idővel kapcsolatos további probléma az, hogy mint mérőszám teljesen géptől és azon belül is operációs rendszertől függő mennyiség. Így különböző gépeken futó LP rendszerek összehasonlítására nem alkalmas (még különféle átszámítások után sem). Az iterációszám ezzel szemben többé-kevésbé géptől független, inkább a megoldó eljárásra jellemző szám.

A megoldás költsége, mint mérőszám azért nem jelent biztos alapot, mert egy bármikor bekövetkezhető tarifális változás egy korábban gazdaságtalanabb eljárást a többi elé tud helyezni anélkül, hogy azon a legcsekélyebb fejlesztés is történne.

A munka mellett a hatékonyság meghatározására szolgáló másik faktor az erőforrás igénybevétel. Ez gyakorlatilag az LP rendszernek a valós memóriaigényét és a szükséges háttértároló kapacitást jelenti. Ha ezt használjuk a hatékonyság mérésére, akkor azt a rendszert tekintjük hatékonyabbnak, amelyiknek kisebb az erőforrás igénye. Ennek a mérőszámnak a jelentősége napjainkban éppen növekszik, mert

— az egyre jobban terjedő kis számítógépek erőforrás ellátottsága korlátozott,
— a nagy gépeken általában külön meg kell fizetni a ténylegesen használt erőforrásokat, illetve nem adnak lehetőséget az összes erőforrás használatára.

Röviden szólni kell egy viszonylag könnyen értelmezhető követelményről, amit egyszerűen a *kapacitás* fogalmával hozhatunk kapcsolatba. Ez azt jelenti, hogy — lehetőleg minél kisebb erőforrás igény mellett — minél nagyobb méretű feladatokat lehessen megoldani. Ahhoz, hogy ennek az igénynek eleget lehessen tenni, nem elég pusztán a program készítése során megfelelően nagy tömböknek helyet biztosítani, hanem bonyolult eljárások sorát kell kialakítani mind az adatkezelés, mind pedig a számítási rész vonatkozásában. Ennek belátásához elég arra gondolnunk, hogy a „tördelt" tömbkezelés mennyivel bonyolultabb, különösen ha ez még tömörített tárolással párosul, illetve a nagyméretű feladatok pontossági követelményei milyen plusz terheket rónak a megoldó eljárásra.

Az LP rendszerekkel szemben támasztott lényeges követelmény a *használati kényelem* (ease of use, user friendliness). A használati kényelem sem egy egzakt fogalom, de elég jól körülírható, hogy mit értünk rajta. Egy felhasználó szempontjából egy LP rendszer felhasználói kényelme nagy, ha igényeit minél teljesebb mértékben és minél egyszerűbben, minél kevesebb emberi munka árán elégíti ki. Ezzel a körülírással egyetérteni viszonylag könnyű, teljesíteni azonban igen nehéz. Elég arra gondolni, hogy mennyire különbözőek lehetnek a felhasználói igények: van aki

azt szeretné, hogy az adatmegadáson túl mással ne kelljen törődnie, a rendszer minden további nélkül, fekete dobozként működve, adja meg a pontos eredményt, és azt is a lehető leghamarabb (esetleg legolcsóbban); más viszont speciális igényeket kíván kielégíteni speciális módon, nagy mértékben egyéni elképzelés szerint használva a rendszer szabad paramétereit, vagy azok egy részét és neki az jelenti a kényelmet, ha erre egyáltalán van — és azon belül pedig minél egyszerűbb — lehetősége. Látható, hogy az első típusú felhasználó lényegében az összes korábban már bemutatott követelmény „automatikus" teljesülését támasztja a rendszerrel szemben. A második típusú felhasználó esete ennél bonyolultabb. Ő ugyanis az eredetileg összehangolt paraméterek közül néhányat megváltoztat és azt várja el, hogy a többi lehetőleg „automatikusan" álljon be olyan értékre, hogy a paraméterek közt ismét olyan összhang legyen, mely képes elősegíteni valamelyik követelmény teljesülését.

Megjegyzendő, hogy az LP rendszerekkel szemben támasztott fenti követelmények más számítástechnikai rendszerek értékelésénél is szerepet játszhatnak.

E helyen célszerű megemlíteni, hogy a szimplex eljárás valójában egy algoritmus család és számítógépes implementációinak paraméterei közt egyaránt vannak kvalitatív és kvantitatív jellegűek. Az alábbiakban — a teljesség igénye nélkül — megemlítünk mindkét csoportból néhány jellemző paramétert.

i) Kvalitatív paraméterek:

— egy optimalizáló algoritmus kiválasztása egy algoritmus családból (pl. primál, duál),
— választás induló eljárásokból (pl. triviális bázis, megadott rész- vagy teljes bázis, CRASH technika [27], ún. „nulladik fázis" [31]),
— választás különféle normálási eljárások közül,
— eljárás választása az első fázisra,
— degenerációs ciklizálás elleni eljárás (pl. perturbáció, lexikografikus eljárás, relaxáció),
— optimalizálási technika (pl. legnegatívabb árnyékár, legnagyobb javítás, devex variánsok [4], [10], [23], többszörös és részleges kiválasztás [27]).

ii) Kvantitatív paraméterek:

— különféle tűrési paraméterek (pl. pivot elem választásánál, árnyékáraknál, abszolút nullázásnál, relatív nullázásnál, megoldás pontosságának ellenőrzésénél, inverz pontosságának ellenőrzésénél, pivot választásnál az invertálás során, fizibilitás ellenőrzésénél, perturbációs technika alkalmazásánál),
— stratégiai paraméterek (pl. összetett [composite] típusú első fázisnál az elsődleges és másodlagos célfüggvény súlyának aránya, a többszörös kiválasztásban résztvevő vektorok maximális száma, minor iterációk maximális száma, részleges kiválasztás „mettől-meddig" paraméterei, első fázisban az infizibilitások súlyozása),
— frekvenciák (pl. bázis újrainvertálásának gyakorisága, megoldás ellenőrzésére, kimentésére, kiíratására vonatkozó gyakoriságok),
— egyéb paraméterek (pl. puffer területek méretei, mágneslemezes munkaterületek elosztása, degenerációs ciklizálás figyelésére vonatkozó paraméter, összes iterációk megengedett száma).

Talán ez a korántsem teljes lista is alátámasztja azt, hogy nem feltétlenül könnyű feladat a paraméterek helyes beállítása. (Kicsit tüzetesebb vizsgálódással az is kiderül, hogy például a tűrési paraméterek megválasztásánál ismerni kell még az adott gép lebegőpontos számábrázolásának relatív pontosságát is.)

A szimplex algoritmusok korábbi számítógépes implementációi nem engedték meg ennyi paraméternek a szabad mozgását, így a felhasználónak kevesebb dolga volt azokkal a programokkal. Igaz, a kapott szolgáltatás színvonala is alacsonyabb volt: a megoldható feladatok méretkorlátai elég alacsonyak voltak, az algoritmusok robusztussága és hatékonysága viszonylag korlátozott volt. Valójában azokban az eljárásokban is benne volt a felsorolt paraméterek nagy része csak rögzítve egy fix értéken és változtathatatlanul. Elsősorban a gyakorlat által felvetett problémák vezettek aztán azokhoz a felismerésekhez, hogy bizonyos helyeken a szimplex módszer eredeti kötöttségei feloldhatók, egyes részek algebrailag ekvivalens alakokkal helyettesíthetők és ezáltal az algoritmus minősége javítható.

A sok paraméter szerepe éppen az, hogy segítségükkel a programot minél jobban hozzá lehessen hangolni az aktuálisan megoldandó feladathoz annak érdekében, hogy azt a lehető legbiztosabban, legpontosabban vagy leggyorsabban (legolcsóbban) oldja meg.

Nyilvánvaló, hogy a paraméterekben megmutatkozó sok lehetőség a helytelen választás veszélyét is magában hordozza, ami pedig azt eredményezheti, hogy az LP rendszer minőségileg és/vagy mennyiségileg hibás eredményt képes szolgáltatni.

*Minőségileg hibás*nak mondjuk az *eredményt,* ha az eljárás

— fizibilis feladatot infizibilisnek talál (esetleg fordítva, ami túl nagy fizibilitási tűrés esetén fordulhat elő),
— véges optimummal rendelkező feladatra azt találja, hogy a megoldás nem korlátos (esetleg fordítva, ami több tűrési paraméter együttesen kedvezőtlen értékéből adódhat),
— olyan bázist deklarál optimálisnak, melyhez tartozó megoldás az optimális megoldástól egy tűrési küszöbértéknél nagyobb mértékben tér el.

Egy *eredmény mennyiségileg hibás,* ha a megoldásban a változók értékei részben, vagy teljesen pontatlanok, vagyis ha kevés, vagy egyetlen pontos jegyet sem tartalmaznak.

A paraméterek kedvező értéke az elérni kívánt céltól és a konkrét feladattól függően változhat, sőt még egy feladaton belül is a megoldás aktuális menetétől függhet (pl. az újrainvertálások gyakorisága).

A felhasználói kényelemnél figyelembe vett első típusú felhasználó tulajdonképpen azt várja el, hogy az összes paraméter egyenként olyan értékre álljon be, melyek összhatásukban a rendszer legkedvezőbb működését eredményezik, míg a második típusúnak az a jó, ha az általa megadott paraméterekhez a többi paraméter úgy igazodik hozzá, hogy az így adódó paraméterrendszer minél jobban feleljen meg a kitűzött célnak. Természetesen lehet olyan felhasználó is, aki éppen kísérletezni óhajt a paraméterrendszerrel és azt kívánja, hogy az általa nem érintett paraméterek ne „igazodjanak" a megadottakhoz, hanem maradjanak meg pl. a standard értékükön.

A paramétereknek a fenti értelemben vett helyes megválasztása igen bonyolult feladat és nehéz egy olyan felügyelő algoritmust megtervezni, mely minden feltétel-

nek eleget tudna tenni: ha kell, valamilyen cél érdekében optimálisan választja meg a paraméterek értékeit, máskor pedig mindent, vagy bizonyos értékeket kienged az ellenőrzése alól. Ennek a problémának a fokozatos megoldására egy kínálkozó lehetőség az adaptivitás elvének az alkalmazása az LP rendszerekben.

3.2. *Az adaptivitás elvének alkalmazása a lineáris programozásban*

3.1. DEFINÍCIÓ: Egy olyan algoritmust, amely az aktuálisan megoldandó feladat ismeretében a paramétereit képes magának beállítani, adaptív algoritmusnak nevezzük.

Ezt a meghatározást nem tekintjük véglegesnek. A paraméterek változtatása ugyanis nem cél nélkül történik. Ezért az LP rendszerek esetében az adaptivitást az alábbiak szerint értelmezzük:

3.2. DEFINÍCIÓ: Egy LP algoritmus adaptív, ha képes a megoldandó feladatot előre és/vagy menet közben elemezni és ennek eredményeként — paramétereinek be-, vagy átállításával — úgy alakítani magát, hogy a végeredményt valamilyen szempontból a legkedvezőbben érje el. Ez a szempont elsősorban a pontosság, megbízhatóság és hatékonyság lehet.

A fenti definíciókban a paraméter fogalmát a szokásosnál általánosabban kell érteni: itt ugyanis nemcsak numerikus jellegű paraméterekről van szó, hanem döntési változókról is. Ennek a részleteit a későbbiekben tárgyaljuk.

A 3.1. definícióra a továbbiakban mint a gyenge értelemben vett adaptivitás, míg 3.2. definícióra mint az erős értelemben vett adaptivitás meghatározására fogunk hivatkozni.

Az adaptív értelemben vett jó döntések hozatala azon múlik, hogy

— találjunk megfelelő kritériumokat az aktuális helyzetnek, illetve a megoldás menetének az ellenőrzésére és
— meg tudjuk határozni (vagy legalább meg tudjuk sejteni, vagy megfelelő indokok alapján el tudjuk várni), hogy mi lesz a hatása a paraméterek változtatásának az adott helyzetben.

Megjegyezzük, hogy ami a zárójelben áll, az bizonyos mértékig a heurisztika területéhez tartozik.

Jelenleg nem ismeretes olyan adaptív felügyelő algoritmus, mely erős értelemben lenne adaptív és képes lenne arra, hogy minden feladatot a kitűzött célnak megfelelően optimálisan oldana meg. Ennek az elvárásnak a fokozatos közelítésére viszont elég jó remény van.

Az LP-ben használt adaptív elemek jellemzője az, hogy nem vezetnek ki a szimplex módszer kereteiből.

Az adaptivitás elvének alkalmazásával nemcsak a paraméterek értékének a beállítását lehet elérni, hanem a „váratlan" helyzetek lekezelését is. Itt természetesen a matematikailag váratlan helyzetekről van szó, mint például amikor egy újrainvertálás nem reprodukálja az invertálás előtti helyzetet. Ilyenkor egy adaptív felügyelő algoritmus a megoldás addigi menetétől függően más és más folytatást, esetleg az eljárás további menetének felfüggesztését választhatja.

Adaptivitással — egyelőre — nem lehet alapvetően új algoritmikus eljárásokat létrehozni a lineáris programozásban. Arra azonban lehet törekedni, hogy az új algoritmikus technikák vagy maguk is adaptívak legyenek, vagy jól beépüljenek a már meglevő adaptív elemek közé, ne rontsák le ok nélkül azok kedvező működési lehetőségét.

Jelenleg az adaptivitástól az várható el, hogy „automatikus", vagyis külső beavatkozás nélküli döntést hozzon az alábbi kérdésekben:

a) bizonyos eljárások illetve technikák használata vagy mellőzése (pl. CRASH, részleges kiválasztás, a szuper ritkássági technika alkalmazása),
b) választás alternatív eljárások között (pl. az inverz szorzat alakja vagy eliminációs alakja; primál vagy duál eljárás illetve lépés választása),
c) adattár felosztása, alkalmazkodás az adott memóriaterülethez (alapmátrixot és az éta file-t a központi memóriában tárolni amíg csak lehet, de ellenkező esetben is csak a túlcsorduló részt tenni a háttértárolóra),
d) a számítási hiba terjedésének szabályozása (pl. a transzformációs képleteknél az eredmény nullázása dinamikus nulla-kritérium alapján),
e) az esetleg felhalmozódó számítási hiba korrekciója (pl. az inverzzel végzett műveletek pontosságának dinamikus kiértékelése és ha szükséges, soronkívüli újrainvertálás kezdeményezése),
f) a tolerancia értékek szabályozása (pl. a toleranciák hozzáillesztése a számok nagyságrendjéhez és/vagy a gépi számábrázolás relatív pontosságához),
g) algoritmikus eljárások numerikus paramétereinek szabályozása (pl. a többszörös kiválasztásban megengedett vektorok maximális száma, lépésszám korlátozása a szuboptimalizálásban, részleges kiválasztás stratégiai paraméterei),
h) a megoldandó feladat átalakítása (pl. alsó korlátok kiiktatása transzformációval, az explicite megadott egyedi felső korlátok kiszűrése, redundáns feltételek ideiglenes törlése, normálás),
i) az optimalizálás menetének szabályozása (pl. árnyékárak dinamikus normálása, dinamikusan súlyozott összetett célfüggvény az első fázisban, degenerációs ciklizálás elkerülése, numerikusan stabilabb lépések preferálása, újrainvertálás szükségességének dinamikus megállapítása),
j) felhasználók által megadott paraméterek ellenőrzése és korrekciója (pl. értelmetlen értékek [negatív normálófaktor, számjegyek helyett betűk, stb.], teljesíthetetlen követelmények támasztása [nemlétező nagyságú memória lekötése, túl sok vektor jelölése a szuboptimalizálásra, stb.], nyilvánvalóan téves értékek korrekciója [az a_{ij}-khez képest irreálisan nagy tűrési paraméter értékek, újrainvertálás elrendelése minden lépésben, stb.]).

Megjegyezzük, hogy a *j)* alatt említett adaptív szolgáltatások megtervezése és azok igénybevétele elég sok szubjektív lehetőséget és elemet tartalmaz és a továbbiakban nem foglalkozunk velük érdemben.

4. LP implementációk főbb problémái

Ebben a fejezetben áttekintjük az LP implementációk főbb problémáit. Ennek során bemutatjuk az LP rendszerek működésének fontosabb jellemzőit, érzékeny pontjait, illetve azokat a szempontokat és feltételeket, melyeket az LP fejlesztő munka során figyelembe kell venni. Ez egyúttal segítséget nyújt ahhoz, hogy összehasonlítani és értékelni lehessen a különböző LP rendszerekben az azonos részfeladatokra kidolgozott megoldásokat, illetve ki lehessen mutatni bizonyos funkciókat ellátó algoritmikus elemek meglétét vagy hiányát. Így lehetőség nyílik arra is, hogy egy-egy adaptív elem beépítésének a hatását módszeresen vizsgáljuk összehasonlítva egy LP rendszer korábbi és új változatát.

4.1. *Számítógépes implementáció során szereplő műveletek áttekintése*

Ennek a pontnak az a célja, hogy rávilágítson arra, mennyire kevés, ha egy algoritmus műveletigényénél csak a lebegőpontos szorzások/osztások számát vesszük figyelembe.

A szimplex iterációk során előforduló fontosabb műveletek és tipikus műveleti idők [30] μsec-ben a következők

a) lebegőpontos aritmetikai műveletek (főleg a szimplex módszer transzformációs képleteinél)

összeadás/kivonás	1,5		
szorzás: szimpla pontosság	4,0	dupla pontosság	5,4
osztás: szimpla pontosság	6,5	dupla pontosság	10,0

b) Fixpontos aritmetikai műveletek (pl. indexkifejezések kiszámítása, ciklusok szervezése, számlálók növelése, adminisztrációk vezetése),

összeadás/kivonás	0,6
szorzás	4,0
osztás	7,5

c) Logikai műveletek (pl. a legkülönfélébb feltételek ellenőrzése, halmazhoz tartozás vizsgálata, ciklusok szervezése),
idők: 0,5—1,5

d) Adatmozgatás a háttértároló és a központi memória közt: input/output (I/O) műveletek. Szekvenciális hozzáférés és 1000 elemű valós típusú blokkok esetén 1 elemre átlagosan 300 μsec (táron belüli hozzáférés 0,3 μsec).

e) Egyéb (pl. egyszerű vezérlés átadás, szubrutin hívás),
idők: erősen eltérőek 0,5—30,0.

A műveleti idők tipikusságán azt kell érteni, hogy a műveleti idők arányai a különböző számítógépeken hasonlóan néznek ki.

A fenti műveletek közül néhányra az igaz, hogy bizonyos esetekben meg lehet őket takarítani (pl. elegendően nagy memória esetén az I/O műveletek egy részét, vagy akár az egészét), vagy másikkal lehet helyettesíteni (pl. logikai művelettel arit-

metikai műveletet, vagy műveleteket lehet megtakarítani, lásd a 2.3.1. lemmát; vagy többlet számítási munkával I/O-t lehet csökkenteni).

A műveleti idők arányaiból az is világosan látszik, hogy az additív és multiplikatív lebegőpontos műveletek idejei összemérhetők, így a műveletek számbavételénél mindegyikre figyelemmel kell lenni. Ez — bár nagyságrendi változást nem okoz — a számítási munkára vonatkozó becslés konstans szorzóját módosítja. Hasonló gondolatmenet érvényes értelemszerűen a többi műveletre is.

A felsorolás egyben támpontot is jelent arra nézve, hogy ahol alternatív megoldási lehetőségek állnak fenn valamilyen részprobléma megoldására, akkor hogyan lehet kiválasztani a gazdaságosabb megoldást, de arra nézve is nyújt segítséget, hogy megállapítsuk, melyek a „drága" részek az algoritmusban, hol érdemes kutatni a megtakarítás lehetőségét.

4.2. *Néhány szó a memóriafelosztásról*

Közismert, hogy a jelenleg használatos számítógépek memóriája egyaránt szolgál a program és az adatmezők tárolására. Ha a rendelkezésre álló memória terület nem elegendő a teljes program és az összes adatmező egyidejű befogadására, akkor szükség van a program és/vagy az adatok átlapolt kezelésére (program-overlay, adat-overlay). Az átlapolási struktúra kialakítása hatással van a program hatékonyságára. A hagyományos operációs rendszerekkel működő gépek esetén a program tervezőjének ebben nagy szabadsága van, míg a virtuális tárkezelő operációs rendszerek esetén ez a lehetőség nagy mértékben csökken.

E helyen kell szólni néhány szót az adattárolási technikákról [8].

*Explicit tárolás*nak azt nevezzük, ha egy vektor, vagy mátrix minden egyes elemét tároljuk valamilyen rendben (mátrix esetén leggyakrabban oszlopfolytonosan). Az alapmátrixnak ilyen formában történő tárolása igen nehézkes és gazdaságtalan lenne, hiszen — mint a 2.2. pontban arról már szó volt — nagyméretű LP feladatok feltételi mátrixa általában nagyon kis kitöltöttségű, ugyanakkor jelen módszerrel az összes 0 elem is tárolódna. Egy valós méretű feladat néhány millió lehetséges elemének tárolása és iterációnkénti mozgatása szinte megoldhatatlan feladat, így az alapmátrix tárolása csak más módon oldható meg. Explicit formában csak a nagy kitöltöttségű vektorokat szokás tárolni.

A (2.1)-beli $a_{ij} \neq 0$ elemek tárolása leggyakrabban *indexes formá*ban történik. Ebben az esetben a nullától különböző elem helymegjelölő adatait (i és j) valamint magát az a_{ij} elemet kell tárolni. Oszlopfolytonos tárolás esetén a helyzet tovább egyszerűsödik, és az oszlop indexet csak egyszer kell megadni, majd pedig oszlopon belül (index, elem) párokat. Ha egy fixpontos index értéket fele akkora helyen tárolunk mint egy lebegőpontos elem értéket, akkor — amint ez könnyen kimutatható — pusztán tárolási szempontból a 67%-nál kisebb kitöltöttség esetén az indexes tárolás gazdaságosabb az explicitnél. Az indexes tárolás többlet adminisztrációs munkája miatt valójában alacsonyabb a határ. Indexesen célszerű tárolni az éta vektorokat is, mert az éta file átlagos sűrűsége mélyen a kritikus érték alatt helyezkedik el.

A KALAN [15] által bevezetett *szuper ritkássági technikán*ak a lényege az, hogy csak az egymástól különböző nem-nulla elemeket tárolja egy táblázatban *(literal pool)* és az oszlopfolytonos indexes tárolásnál használt (index, elem) párt (index_1,

$index_2$) párral helyettesíti, ahol a második index az a_{ij} elem táblázatbeli helyét mutatja meg. A táblázat egyszeri felépítése után ez a technika további tárigény csökkenést és ezzel kapcsolatos sebességnövekedést eredményez — további többlet adminisztrációs munka árán.

A számítógépes hardware fejlődésének egy új vívmánya kissé a háttérbe szorítja a szuper ritkássági technikát. Arról van szó, hogy kifejlesztettek 64 Kbyte nagyságú szuper gyors memóriamodulokat, melyeken belül a műveletek 1 nagyságrenddel gyorsabban végezhetők el, mint a mellettük működő hagyományos memóriában. Ha azonban az említett literal pool túl nagy és minden elem hozzáféréséhez ki kell nyúlni a szuper gyors memóriából, akkor elvész a gyors memória majdnem minden előnye. Ellenben ha szekvenciálisan, az indexes tárolásnak megfelelő formában nagyobb részeket egyszerre hozunk be és utána végig a 64 K-n belül működik a rendszer, akkor a gyorsaság nagy mértékben realizálható. Ilyen megoldásra a szuper ritkássági technikánál azért nincs lehetőség, mert a literal pool egyes pozícióihoz a hozzáfordulás véletlenszerű *(random)*.

A 64 Kbyte-os szuper gyors memória előnyeinek a kihasználásánál előtérbe kerülnek a korszerű kisgépes LP technikák.

A központi memóriában biztosított adatmezőn sok és különféle információt kell tárolni. Az információk egy része adminisztratív jellegű, más része pedig a szimplex iterációk numerikus képleteivel kapcsolatos adatokat hordozza. Tipikus az az eset (és az LP rendszereket erre gondosan fel kell készíteni), hogy az összes információ nem fér el a memóriában. Ilyenkor az adatmezőt fel kell osztani kisebb részekre a különféle információk közt és így mindegyikből annyi adat lesz közvetlenül hozzáférhető, amennyi a számára a központi memóriában kijelölt helyen egyszerre elfér. Ez azt jelenti, hogy átlapolt adattárolási és hozzáférési (adat-overlay) technikát kell alkalmazni, aminek megvalósítása igen gondos mérlegelést igényel, mert jelentős hatással van az LP rendszerek hatékonyságára.

A rendelkezésre álló memória jó kihasználása *dinamikus memória felosztás*t követel meg. Ez azt jelenti, hogy ugyanazon a helyen más és másfajta információt tárolunk az aktuális helyzettől függően. Az adminisztratív jellegű információ zöme a bázisra vonatkozik (bázisváltozók indexei, típusa, fizibilitási állapota, stb.) és fixpontos változókban tárolható egyenként egy-egy m komponensű vektorban. További numerikus adatok ugyancsak a bázisra vonatkoznak (bázisváltozók felső korlátai, aktuális bázismegoldás, szimplex szorzó, stb.) és tárolásukhoz ugyancsak egy-egy m komponensű, de most lebegőpontos elemekből álló vektor kell. Ezek részletes ismeretére a továbbiakhoz nincs szükség. Röviden vázolni kell azonban három egyéb adatfajta tárolási rendjét. Ezek: a feltételi mátrix adatai, az éta vektorok és a szuboptimalizálásban résztvevő vektorok. Az első kettő legtöbbször indexes formában tárolódik, míg a harmadik explicit alakban. Mindhárom adatcsoportra egyszerre nincs szükség. A 2.3. pontban ismertetett RSM2 algoritmus esetében a három kiemelt adatfajtára vonatkozó helyzet az iterációs lépések során a következő:

— BTRAN-hoz csak az éta vektorokra van szükség, valamint a szimplex szorzóhoz szükséges, (2.16)-ban definiált $\mathbf{v}$ vektorra.
— MPRICE először csak a feltételi mátrixot használja, majd ezután előveszi és explicit módon tárolja a szuboptimalizálásban résztvevő k vektort.
— MFTRAN az éta vektorokkal transzformálja a kiválasztott k vektort.
— PIVOT és TRANSFORM a kiválasztott vektorokkal dolgozik.

Ebből rögtön látszik, hogy a kiválasztott vektorok területére, amit az $\alpha = \mathbf{B}^{-1}\mathbf{a}$ típusú adatok tárolása miatt α területnek nevezünk, nincs szükség a BTRAN és az MPRICE első része során, így azok helye rendelkezésre áll az éta vektorok illetve a feltételi mátrix számára. MFTRAN bizonyos mértékig szűk keresztmetszetet jelent, mert az éta vektorok mellett a kiválasztott vektorokra is szüksége van, így most kevesebb hely jut az étáknak. A PIVOT és a TRANSFORM során azonban az éták helye felszabadul.

4.3. *Implementációs alapproblémák*

A szimplex módszer egy olyan iterációs eljárás, mely alapvetően lebegőpontos számítási műveletekkel dolgozik. A megoldás eléréséhez szükséges iterációs lépések száma előre nem ismeretes és a lebegőpontos számokkal végzett műveletek halmozódó számítási hibákkal lehetnek terhesek. Mindezek együtt különféle algoritmikus és numerikus problémákat vetnek fel. Mindezekhez járul a mai korszerű, de egyben bonyolult LP rendszerek strukturális problémája. Jelen pontban ezeket a problémákat tekintjük át.

a) Algoritmikus problémák

Noha a szimplex eljárás elméletileg véges, a gyakorlatban mégis gond, hogy a megoldás eléréséhez nagy számítási munkát kell elvégezni. Ez egyrészt azért van, mert az eljárás *konvergenciája* általában *lassú* (vagyis az optimális bázisra való ráinvertáláshoz képest sok iterációs lépésre lehet szükség), másrészt pedig *sok* lehet *az egy iterációra eső számítási munka.*

Az iterációk számának csökkentésére tett erőfeszítések általában együtt járnak az egy iterációra eső munkának a növekedésével, így ez a két tényező egymás ellen dolgozik. Nem könnyű megtalálni azt a pontot, hogy meddig érdemes elmenni a munka növelésével, mert míg a ráfordítás növekedést többé-kevésbé jól lehet mérni, az iterációk számában bekövetkező várható csökkenést már jóval nehezebb megbecsülni. Ennek a kérdésnek a közelítő megválaszolására szolgál a *teljesítményvizsgálat (benchmark)*, ami abból áll, hogy egy gondosan összeválogatott feladathalmaz (tesztfeladattár) elemeit oldjuk meg az összehasonlítani kívánt algoritmusokkal és a teljesítményi mutatókat (iterációszám, CPU idő, csatorna idő, stb.) kiértékeljük.

Az algoritmikus problémák megoldásában jelentős szerepet játszhatnak az adaptív elemek.

b) Numerikus problémák

A numerikus problémák alapvetően abból fakadnak, hogy lebegőpontos (valós típusú) mennyiségekkel végzünk el nagyszámú aritmetikai műveletet. Tekintettel arra, hogy a valós típusú számok tárolása és velük a műveletek elvégzése véges pontossággal történik, a műveletek eredményében *jegyveszteségi* és *kerekítési hibák* egyaránt lehetnek. A multiplikatív műveleteknél a kerekítési hiba fordulhat elő, míg az additív műveleteknél mindkettő.

Numerikus szempontból az additív műveletek a veszélyesebbek, mert:

i) Nagy lehet a relatív hiba [18], [29] a magas helyiértékű jegyek elvesztése esetén.

ii) Az additív műveletek a számítógépen nem asszociatívak! Ezt legjobban egy példával lehet illusztrálni. Tegyük fel, hogy a z_1, z_2 és z_3 számokat össze kell adni. Ötjegyű decimális lebegőpontos ábrázolást és duplapontosságú műveletvégzést feltételezve legyen $z_1 = 0{,}99313 \cdot 10^{-2}$, $z_2 = 0{,}75542 \cdot 10^{3}$, $z_3 = -0{,}75543 \cdot 10^{3}$. Könnyen belátható, hogy az összeadásokat különböző sorrendben elvégezve más-más eredményt kapunk:

$$(z_1 + z_2) + z_3 = 0{,}000\,00 \cdot 10^{0}$$

$$z_1 + (z_2 + z_3) = 0{,}687\,00 \cdot 10^{-4}.$$

Egy másik példa azt mutatja, hogy — pl. rossz skálázás miatt fellépő — nagyon eltérő nagyságrendű számokkal végzett additív műveleteknél mit tud eredményezni az asszociativitás hiánya:

ha V egy olyan „nagy" szám, x-hez képest, hogy

$$|V| \cdot \varepsilon > |x|,$$

ahol ε a gépi számábrázolás relatív pontossága, akkor

$$(x + V) - V = 0$$

adódik az algebrailag helyes x helyett, mert a zárójelben álló mennyiség értéke a véges számábrázolás miatt V lesz. A példa értelemszerűen általánosítható és könnyen adódik a következtetés, hogy a skalár szorzat az egyik legveszélyesebb művelet. Ennek használata azonban az LP rendszereknél elkerülhetetlen: pl. árnyékárak meghatározása (2.11) alapján illetve a BTRAN művelet (2.18) szerint. Természetesen a (2.20) típusú FTRAN transzformációs procedurában is vannak additív lebegőpontos aritmetikai műveletek.

Itt kell megemlíteni két — numerikus szempontból — veszélyes szokást. Az első a felhasználói gyakorlatban szokott előfordulni, míg a második az LP implementálók körében.

i) Egy $\mathbf{a}_i\mathbf{x} \leqq b_i$ típusú feltétel „felszabadítása" $\mathbf{a}_i\mathbf{x} \leqq V$ formában, ahol V egy kellően nagy pozitív szám. Ekkor a megoldás transzformációja (2.10a) alapján (az egyszerűség kedvéért $i=2$ választással):

$$\begin{bmatrix} 1 & & \eta_1 & & \\ & 1 & \eta_2 & & \\ & & \vdots & & \\ & & \eta_r & & \\ & & \vdots & & \\ & & \eta_m & & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ V \\ \vdots \\ b_r \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} b_1 + b_r\eta_1 \\ V + b_r\eta_2 \\ \vdots \\ b_r\eta_r \\ \vdots \\ b_m + b_r\eta_m \end{bmatrix}.$$

Az eredményvektor második komponensének számított értéke — az előbb említett numerikus okok miatt — V lesz és a hasonló típusú iterációk során

ez az érték marad is. Az igazi probléma akkor jelentkezik, amikor a felszabadított soron történik pivotolás. Ekkor ugyanis ($r=2$ választással)

$$\begin{bmatrix} 1 & \eta_1 & & & \\ & \eta_2 & & & \\ & \eta_3 & 1 & & \\ & \vdots & & \ddots & \\ & \eta_m & & & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ V \\ \beta_3 \\ \vdots \\ \beta_m \end{bmatrix} = \begin{bmatrix} \beta_1 + V\eta_1 \\ V\eta_2 \\ \beta_3 + V\eta_3 \\ \vdots \\ \beta_m \quad V\eta_m \end{bmatrix}$$

tehát az aktuális megoldásban V-hez közeli nagyságrendű számok kerülnek minden olyan helyre, ahol $\eta_i \neq 0$. Egy ilyen pivot választás a második fázisban azt jelentené, hogy a megoldás valójában nem korlátos, de mégis az adódik, hogy a kilépő elem meghatározásánál van egy figyelembe veendő hányados, éppen V/α_i. Az így meghatározott soron pivotolva pedig létrejön a bemutatott helyzet és miután nem biztos, hogy teljesül a leállási feltétel, az iterációs eljárás tovább mehet a teljesen hibás jobboldal értékekkel ami aztán egy értékelhetetlen végeredményhez tud vezetni.

A feltételek ilyen módon történő kiiktatása tehát mindenképpen kerülendő. Vagy ténylegesen törölni kell a feltételt, vagy — ami a 2.1. pontban leírtak alapján nyilvánvaló — a feltételt nem-korlátozónak kell deklarálni, vagyis a logikai változójának 3-as típuskódot kell adni. Így a feltétel kiiktatása numerikus út helyett adminisztratív úton történik meg, ugyanakkor a feltétel kiértékelése is biztosítva van.

ii) Egyedi felső korlátok normálása 1-re, illetve valamilyen más közös értékre. Ez az ötlet elsősorban a megoldandó algoritmus szempontjából jelent egyszerűsítést. Ekkor ugyanis elegendő egy változóról azt az információt tárolni, hogy felsőkorlátos vagy sem, mert ha igen, akkor értéke automatikusan 1, amit tárolni sem kell. Ily módon potenciálisan $2m+n$ lebegőpontos (összes változókra + külön a bázisváltozókra vonatkozó egyedi felsőkorlátok) hely szabadítható fel. Ezért az egyszerűsítésért azonban nagy árat lehet fizetni robusztusságban! Ilyen esetben ugyanis egy $x_j \leqq u_j$ $(u_j \neq 1)$ felsőkorlátot úgy normálunk 1-re, hogy bevezetjük az

$$x_j = u_j \hat{x}_j \tag{4.1}$$

változót, aminek korlátja $\hat{x}_j \leqq 1$ lesz. Ekkor azonban a j-edik oszlop minden együtthatóját is meg kell szorozni u_j-vel:

$$\hat{a}_{ij} = u_j a_{ij}. \tag{4.2}$$

Ha u_j-nek értéke 1-től nagyságrendileg nagyon eltér, akkor ezáltal egy nagyon kiegyensúlyozatlan feltételi mátrixot hozunk létre, melyben az egymással műveleti kapcsolatba lépő elemek nagyságrendben nagyon különbözhetnek egymástól és az említett jegyveszteségi hibákra nagyon hajlamossá teszik a feladatot. Ilyen kezelés esetén aztán különösen veszélyes lenne egy egyedi korlát felszabadítása oly módon, hogy a felső korlátnak valami $V \gg u_j$ értéket adunk, mert ekkor az oszlopelemek Va_{ij} értékűek lesznek.

A numerikus problémák mindenképpen rontanak az LP rendszerek teljesítményi mutatóin. A felmerülő és halmozódó számítási hibák ugyanis a következőkhöz vezethetnek:

— többlet számítási és I/O művelet, mert az újrainvertálások nem reprodukálják az invertálás előtti állapotot, emiatt visszaesések következnek be, amiket újabb munka ráfordításával kell ledolgozni; ha ezek után adódik is helyes végeredmény, akkor is jelentős *hatékonyság romlás* következett be,
— pontatlan (mennyiségileg hibás) eredmény, ami a robusztusság egyik összetevője, a *pontosság romlását* eredményezi,
— téves (minőségileg hibás) eredmény, ami pedig a robusztusság másik összetevőjének, a *megbízhatóságnak a romlását* jelenti.

A numerikus problémák megoldásában ugyancsak jelentős szerepet játszhatnak az adaptív elemek [21], [22].

c) Strukturális problémák

Az LP programcsomagok általában nem egyszer s mindenkorra lezárt rendszerek, hanem — különféle okok miatt — gyakran változnak. A változás célja minden esetben a rendszer jobbá tétele, esetleges hibáinak kijavítása, „elavult" elemeinek újakkal való kicserélése és képességeinek bővítése (módosítás, fejlesztés). Ilyen jellegű változtatások hibátlan és könnyű megtételére akkor van lehetőség, ha maga a programrendszer felépítése kellően moduláris és strukturált. Ez megfelelő, de nem túlzott tagoltságot is jelent egyben. (Túlzott tagoltság esetén túl sokszor lenne szükség a viszonylag lassú szubrutinhívási műveletekre.) Ilyen esetben a programrendszer elkészítése és bejátszása is lényegesen nagyobb hatásfokkal végezhető el és könnyebb a programkövetés is.

Szükséges, hogy a program minden logikailag lehetséges esetre fel legyen készítve és ne tudjon definiálatlan akciókba kezdeni. Ez egyrészt algoritmikus követelmény, de másrészt szerkezeti is, miszerint nem áttekinthető szerkezet esetén ennek a követelménynek a teljesülését nehéz biztosítani [20].

Ugyancsak szerkezeti problémát jelent a külső hozzáférési lehetőség biztosítása. Ezen az értendő, hogy az LP rendszer bizonyos szerkezeti elemeire (pl. BTRAN, FTRAN, PIVOT, INVERT a 2.3-beli RSM2 algoritmusból) a felhasználóknak külön-külön is szükségük lehet különleges igények kielégítésére. Az ilyen elemek autonóm használatra való felkészítése különös gondosságot igényel, viszont sikeres megvalósítás esetén jelentősen növeli a rendszer flexibilitását.

A szerkezeti problémák elsősorban a rendszer megtervezésének a szintjén jelentkeznek és megoldásukban — jelenlegi elképzeléseink szerint — az adaptivitásnak nem jut szerep, hiszen az előre kialakított struktúra működés közben változatlan marad, a program strukturálisan nem változtatja magát.

5. Adaptív elemek a LIPROS-ban

A napjainkban ismert nagy LP rendszerek már tartalmaznak adaptív elemeket. Ezek egy része viszonylag egyszerű (pl. számítási „szemetek" nullázása), mások jóval bonyolultabbak (pl. lexikografikus eljárás beléptetése a célfüggvény hosszú idejű mozdulatlansága esetén).

Jelen fejezet célja az, hogy ismertessen néhány, a LIPROS számára kidolgozott, már beépített illetve tervbe vett és lényegesnek tartott adaptív elemet. Ezen keresztül lehetőség nyílik az adaptivitás jelentőségének szemléltetésére, illetve annak a bemutatására, hogy az eddig tárgyalt elvek milyen módon érvényesülnek új algoritmikus elemek kidolgozásánál. Azáltal, hogy a kidolgozott adaptív elemeket azonnal be lehetett építeni egy egyébként működő LP rendszerbe, lehetőség nyílt azok hatásának közvetlen tanulmányozására. A számítástechnikai tapasztalatok — melyek közül néhányról külön is szó lesz — részben a további fejlesztő munka és azon keresztül további eredmények forrásává váltak.

Az ismertetésre kerülő adaptív algoritmikus elemek a PIVOT, MPRICE és CRASH (ennek értelmezése az 5.3. pontban szerepel) műveletekre terjednek ki és egyrészt az első fázis hatékonyabbá tételére, másrészt a degeneráció káros következményeinek (konvergencia lassítása, hatékonyság romlása) a csökkentésére szolgálnak. Az első fázis — mint szükséges rossz — rövidítésének problémája régóta ismert. A degeneráció pedig azért érdemel komoly figyelmet, mert a gyakorlatban előforduló nagyobb méretű feladatok megoldása során a degeneráció nagyon gyakran és nagy mértékben jelentkezik és bár algoritmikus ciklizálás nem fordul elő, a megoldás menete rendkívül le tud lassulni. Megfelelően felépített adaptív elemek az ilyen helyzeteket automatikusan észreveszik és lekezelik — amennyire ez képességeikből telik —, így a felhasználónak nem kell beavatkozni, hogy átsegítse az eljárást egy-egy ilyen holtponton.

5.1. *PIVOT*

A pivot lépés feladata az, hogy egy adott belépő vektorhoz meghatározza a bázisból kilépő vektort, vegye észre, ha nincs báziscsere, csak korlátcsere, illetve jelezze az esetleges nem-korlátosságot. A továbbiakban kizárólag az első fázisról lesz szó, ahol sikerült egy olyan új eljárást találni, amely nagyon sok tekintetben adaptív módon működik és nagyon kedvező tulajdonságokkal rendelkezik.

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

A hagyományos első fázis eljárások (pl. [9], [28]) 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 (az infizibilitások száma nem növekszik);
iii) a bázisból kilépő vektor fizibilis értéken hagyja el a bázist (alsó vagy felső korláton).

Még ez a három feltétel sem határozza meg egyértelműen a bázisból kilépő vektort és ennek megfelelően az első fázis eljárások több variációját dolgozták ki. Ha relaxáljuk az ii) feltételt, még további lehetőségek kínálják magukat. Jelen pontban egy ennek a kihasználásán alapuló új eljárás bemutatása szerepel. Ez az eljárás minden iteráció során maximalizálja a fizibilitás irányába tett előrehaladást (ez egy nem-

lineáris maximalizálási feladat lesz), amelyet egy adott javító vektorral egyáltalán el lehet érni. Mindez a bázisból kilépő vektor meghatározására szolgáló új szabály segítségével történik, melynek a hagyományossal szemben alig van számítási többlet munkaigénye.

A kilépő vektor meghatározásának ezen elve szükségessé tette egy hozzá jobban igazodó, új oszlopkiválasztási eljárás kidolgozását, mely a belépő vektor meghatározására szolgál. Ez a technika dinamikusan képes figyelembe venni a valódi célfüggvényt az első fázisban. Noha ez az eljárás már valamivel több számítási többlet munkát igényel major iterációnként, mégis elfogadhatónak látszik, mert az iterációk számának és az össz számítási munkának jelentős csökkenését tudja eredményezni.

A bemutatandó eljárások implementációját elkészítettem, amelyek ily módon beépültek a LIPROS-ba. Az eddig szerzett számítástechnikai tapasztalatok igen kedvezőek, de természetesen szükség van még további tesztelésre.

A teljesítmény jelentős javulása elsősorban az alábbi tulajdonságoknak köszönhető:

— az infizibilitás mértékének sokkal nagyobbfokú csökkentése,
— degeneráció esetén lényegesen hatékonyabb lépések,
— megnövekedett numerikus stabilitás,
— általában igen kevés lépés marad a második fázisra.

Ez utóbbi azt jelenti, hogy a feladat általában nagyrészt megoldható az első fázisban, az össz számítási munka jelentős csökkenése mellett.

A kilépő vektor meghatározására szolgáló eljárásra a továbbiakban DELPHI néven fogunk hivatkozni. Az adaptív oszlopválasztás neve ADACOMP. Ebben a pontban a DELPHI eljárás tárgyalása szerepel, míg az ADACOMP részleteiről az 5.2. pontban lesz szó.

5.1.1. *A DELPHI eljárás*

A (2.5), (2.6)-tal felírt feladat jelöléseit az alábbiakkal egészítjük ki:

I_B bázisváltozók indexhalmaza

I_0 0 típusú bázisváltozók indexhalmaza

I_1 1 típusú bázisváltozók indexhalmaza

I_2 2 típusú bázisváltozók indexhalmaza

I_3 3 típusú bázisváltozók indexhalmaza

Nyilvánvalóan:

$$I_B = I_0 \cup I_1 \cup I_2 \cup I_3.$$

Ha egy x_j változó 0 típusú, akkor felső korlátját $u_j = 0$-val definiáljuk és a felső korlátos bázisváltozók indexhalmazát I_u-val jelöljük:

$$I_u = I_0 \cup I_1.$$

A bázisváltozókat fizibilitási állapotuk alapján három csoportba osztjuk (u_i-vel jelölve az i-edik bázisváltozó felső korlátját)

$$M = \{i\colon i \notin I_3 \wedge \beta_i < 0\}$$
$$P = \{i\colon i \in I_u \wedge \beta_i > u_i\} \tag{5.1}$$
$$F = I_B \setminus (M \cup P)$$

M tehát azon bázisváltozók indexhalmaza, melyek Mínusz irányban infizibilisek. P pedig azoké, melyek Plusz irányban infizibilisek. F a Fizibilis értéken levő bázisváltozók indexhalmaza. Egy 3 típusú bázis változó mindig fizibilis, így mindig F-hez tartozik.

Egy bázismegoldás infizibilitásának a mértékét a következőképpen definiáljuk:

$$w = \sum_{i \in M} \beta_i - \sum_{i \in P} (\beta_i - u_i). \tag{5.2}$$

Ez a definíció hasonlatosságot mutat az ORCHARD-HAYS [27] által használt definícióhoz,

$$w = \sum_{i \in M} \beta_i - \sum_{i \in P} \beta_i \tag{5.2a}$$

de attól érdemben eltér, amint az a későbbiek során kiderül. (5.2)-t úgy lehet értelmezni, mint az infizibilis bázisváltozók fizibilitási tartományuktól mért távolságösszegének -1-szerese. Nyilvánvaló, hogy

$$w \leqq 0 \tag{5.3}$$

és csak akkor lesz $w=0$, ha mind az M mind a P halmazok üresek, vagyis a megoldás fizibilis. Ennek megfelelően a szimplex módszer az első fázisban arra törekszik, hogy megoldja az alábbi feladatot:

(W) $\max w$
a (2.5) és (2.6) feltételek mellett.

A (W) feladat megoldására a szimplex módszer alapvető technikáját lehet használni, noha ez nem egy szokásos értelemben vett LP feladat, mivel az M és P halmazok változásával a célfüggvény összetétele is változik.

Egyelőre még fenntartjuk az ii) feltételt és tegyük fel, hogy egy x_j bázison kívüli változó 0 szinten van. Azt vizsgáljuk, hogy x_j növelése — melyet t-vel paraméterezünk — milyen hatással van w-re. A (2.6a) alap egyenlőségek fennállásához a bázisváltozók értéke is megváltozik t függvényében, amit az $\mathbf{f}(t)$ vektorral jelölünk. Így az

$$\mathbf{B}\mathbf{f}(t) + t\mathbf{a}_j = \mathbf{b} \tag{5.4}$$

vagy

$$\mathbf{f}(t) = \boldsymbol{\beta} - t\boldsymbol{\alpha}_j \tag{5.5}$$

alakot kapjuk, ahol $\boldsymbol{\alpha}_j = \mathbf{B}^{-1}\mathbf{a}_j$. (5.4)-ben és (5.5)-ben — az egyszerűbb jelölés kedvéért — nem tüntettük fel a baloldalnak a j-től való függését.

(5.5)-öt koordináta alakban is felírjuk:

$$f_i(t) = \beta_i - t\alpha_{ij}, \quad i = 1, \ldots, m. \tag{5.6}$$

Tegyük fel, hogy egy olyan bázisnál vagyunk, melyre $w<0$. Ekkor igaz a következő:

5.1. LEMMA. Ahhoz, hogy egy 0 szinten levő bázison kívüli változó értékének a növelése javítani tudja w-t, szükséges, hogy teljesüljön rá az

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

egyenlőtlenség.

Bizonyítás: Tegyük fel, hogy $t>0$ elég kicsi ahhoz, hogy M és P halmazok változatlanok maradjanak. Ekkor w megváltozása:

$$\Delta w = \sum_{i \in M} [f_i(t) - f_i(0)] - \sum_{i \in P} \{[f_i(t) - u_i] - [f_i(0) - u_i]\} = \sum_{i \in M} (-t\alpha_{ij}) - \sum_{i \in P} (-t\alpha_{ij}) =$$

$$= -t\left(\sum_{i \in M} \alpha_{ij} - \sum_{i \in P} \alpha_{ij}\right) = -td_j.$$

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

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

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

Egy iteráció első lépése az első fázisban tehát abból áll, hogy minden bázison kívüli és nem 0 típusú változóra megvizsgáljuk a d_j értéket és eldöntjük, hogy valamilyen irányban elmozdulva javíthat-e az infizibilitások w mértékén. Több potenciális javító vektor esetén — további szempontok alapján — kiválasztunk egyet, vagy néhányat (MPRICE). Ez a kiválasztás jelentősen befolyásolhatja a megoldás eléréséhez szükséges lépések számát. Ennek részleteiről az ADACOMP tárgyalásánál lesz szó.

A továbbiakban feltesszük, hogy valamilyen módon kiválasztottuk a belépő változót és azt vizsgáljuk, hogy hogyan alakul a bázisváltozók fizibilitási állapota t függvényében az (5.4) egyenlőség teljesülése mellett. A bázisváltozók mozgására tett ii) megkötést mostantól feloldjuk és így a bázisváltozók szabadon mozoghatnak a transzformáció során. A kilépő változó meghatározása azonban még több meggondolást igényel.

A képletek további egyszerűsítése érdekében (5.6) képlet jobboldaláról is elhagyjuk a j indexet és az

$$f_i(t) = \beta_i - t\alpha_i \tag{5.7}$$

alakot tekintjük.

Miután a 3-as típusú bázisváltozók mindig fizibilis értéken vannak, ezért a vizsgálatokból azokat kizárjuk és csak az

$$I = I_B \setminus I_3$$

indexhalmazra szorítkozunk.

Egy I-beli változó negatív irányban infizibilis, ha értéke negatív és egy I_u-beli pozitív irányban infizibilis, ha értéke nagyobb, mint a felső korlátja. Használjuk a

következő jelölést:

$$(5.8)\qquad Z^- = \begin{cases} 0, & \text{ha} \quad Z \geqq 0 \\ Z, & \text{ha} \quad Z < 0 \end{cases}$$

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

Ennek segítségével az infizibilitás mértéke t függvényében az alábbi alakban írható fel:

$$(5.9)\qquad w(t) = \sum_{i \in I} [f_i(t)]^- - \sum_{i \in I_u} [f_i(t) - u_i]^+ =$$

$$= \sum_{i=I} [\beta_i - t\alpha_i]^- - \sum_{i \in I_u} [\beta_i - t\alpha_i - u_i]^+$$

$w(t)$-nek néhány egyszerű tulajdonságát a következő lemma jellemzi:

5.2. LEMMA. $w(t)$ egy folytonos lineáris törtvonalfüggvény, töréspontja ott van, ahol valamelyik változó fizibilitási állapotában változás történik.

A bizonyítás triviálisan adódik (5.8)-ból és (5.9) második sorából. Az is nyilvánvaló, hogy az (5.2)-ben definiált w értéket éppen $w(0)$ szolgáltatja. Ilyen értelemben $w(t)$ az (5.2)-beli w függvényszerű kiterjesztésének tekinthető. Ha erre a célra a [27]-ben szereplő és (5.2a)-ban idézett w-t használnánk, akkor egy nem folytonos függvényt kapnánk, ugyanis az $i \in I_u$ változók hozzájárulása $w(t)$ második szummájához egy u_i értékű ugrást jelentene a korlát átlépésekor, így ez nem lenne alkalmas a további vizsgálatok elvégzésére.

(5.9)-ben egy $f_i(t)$-nek mindaddig van nullától különböző hozzájárulása az első szummához, amíg t értéke olyan, hogy $f_i(t) < 0$, vagyis $i \in M$. Hasonlóképpen $i \in I_u$ esetén $f_i(t) - u_i$ addig járul hozzá a második szummához, amíg $i \in P$. Egy bázisváltozó fizibilitási állapotában akkor következik be változás, amikor eléri fizibilitási tartományának valamelyik határát.

Miután $w(t)$-t a nem-negatív t értékekre vizsgáljuk, ezért a töréspontokat (5.7) alapján az alábbiak határozzák meg:

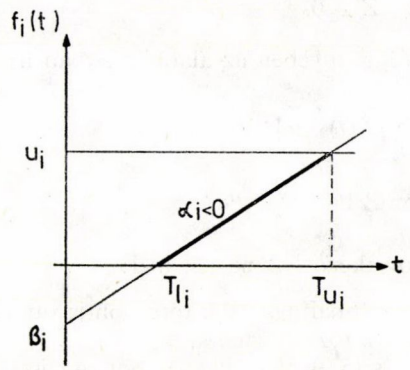
$$(5.10)\qquad \left.\begin{array}{l} T_{l_i} = \beta_i/\alpha_i > 0 \quad (\alpha_i \neq 0) \\ \text{vagy} \quad T_{l_i} = 0, \quad \text{ha} \quad \beta_i = 0 \quad \text{és} \quad \alpha_i > 0 \end{array}\right\} i \in I$$

$$(5.11)\qquad \left.\begin{array}{l} T_{u_i} = (\beta_i - u_i)/\alpha_i > 0 \quad (\alpha_i \neq 0) \\ \text{vagy} \quad T_{u_i} = 0, \quad \text{ha} \quad \beta_i = u_i \quad \text{és} \quad \alpha_i < 0 \end{array}\right\} i \in I_u$$

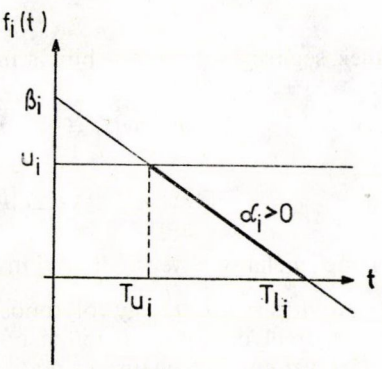
T_{l_i}-vel jelöltük azt a pontot, ahol az i-edik bázisváltozó eléri az alsó korlátját (vagyis 0-t), míg T_{u_i}-vel azt a pontot, ahol az i-edik bázisváltozó eléri felső korlátját, ha ilyen van. A $T_{l_i} = 0$ eset csak akkor határpont, ha infinitézimális elmozdulás esetén az i-edik bázisváltozó negatívvá válik, vagyis átkerül az M halmazba. Hasonlóképpen a $T_{u_i} = 0$ eset csak akkor határpont, ha infinitézimális elmozdulás esetén az i-edik bázisváltozó a felső korlátjánál nagyobb lesz. Az $\alpha_i = 0$ eset jelen szempontból érdektelen, mivel egy ilyen i-hez tartozó bázisváltozó nem mozdul el, így fizibilitási állapota sem változik.

Nyilvánvaló, hogy ha $u_i=0$, akkor $T_{l_i}=T_{u_i}$. Természetesen egyéb módon is lehetnek az (5.10) és (5.11) által definiált küszöbértékek közt egyenlők, így a töréspontoknak lehet „multiplicitása".

A 2. és 3. ábrán példát mutatunk arra, hogy a küszöbértékek hogyan definiálódnak.



2. ábra



3. ábra

A továbbiakban az a célunk, hogy t-nek olyan értéket adjunk, mely maximalizálja $w(t)$-t. Ehhez azonban szükség van $w(t)$ alaposabb ismeretére. Ki fogjuk mutatni, hogy $w(t)$-nek van néhány kedvező tulajdonsága, és a maximalizálást, mely egyben globális maximumot ad, könnyen végre lehet hajtani, továbbá, hogy a maximum egy szimplex típusú báziscserét határoz meg.

Először is emlékeztetni kell arra, hogy az i-edik bázisváltozó, mint a t függvénye

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

alakú. Ebből látszik, hogy az i-edik bázisváltozó fizibilitása függ a típusától, β_i fizibilitásától, valamint α_i előjelétől. $f_i(t)$ hozzájárulását $w(t)$-hez — figyelembe véve $w(t)$ (5.9)-beli definícióját — a 4. ábrán (az $\alpha_i>0$ eset) és az 5. ábrán (az $\alpha_i<0$ eset) lehet szemléltetni. Az ábrákon $f_i(t)$-t vékony vonal, míg $f_i(t)$ hozzájárulását $w(t)$-hez vastag vonal jelöli. A szereplő T_{l_i} és T_{u_i} értékek az (5.10) és (5.11) definícióbeli küszöbértékeket jelentik.

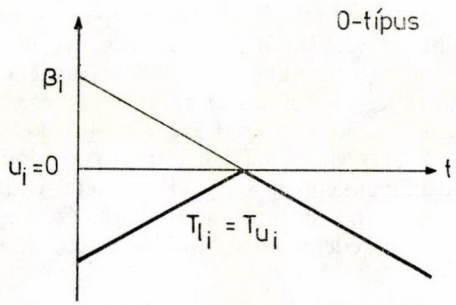
Minden más eset az ábrákon szereplő helyzetek speciális eseteként származtatható.

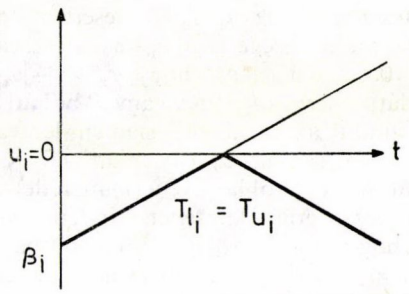
$f_i(t)$ hozzájárulása $w(t)$-hez tehát minden esetben egy konkáv függvény. Miután $w(t)$ éppen ezen konkáv függvények összege, ezért maga is konkáv.

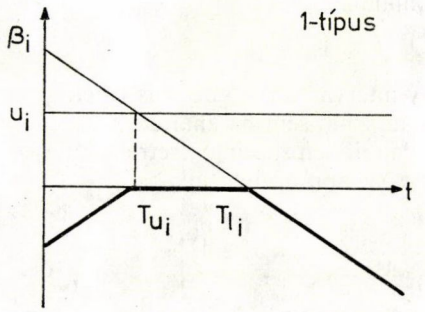
Ezek után rátérhetünk $w(t)$ részletes vizsgálatára. Elindulva a $t=0$-ból, az első töréspont ott lesz, ahol t eléri az (5.10) és (5.11) által definiált küszöbértékek közül a legkisebbet. Szükség van tehát ezen értékeknek nagyság szerint növekvő sorrendbe történő rendezésére:

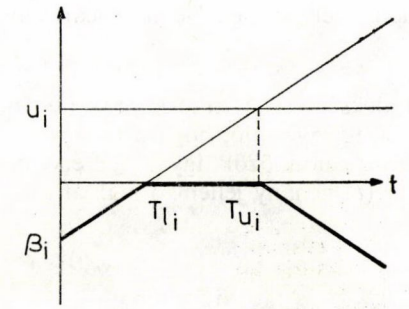
$$0 \leqq t_1 \leqq \ldots \leqq t_Q, \tag{5.12}$$

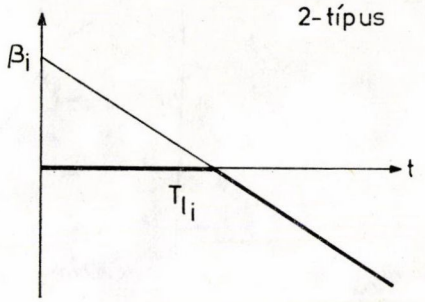
ahol Q jelöli az összes definiált értékek számát.

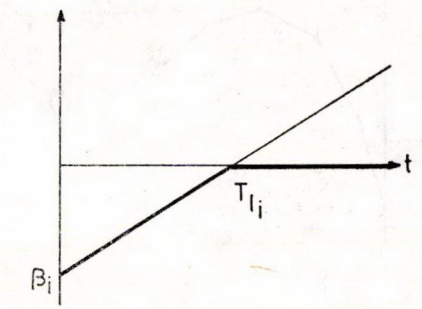












4. ábra. $\alpha_i > 0$ eset

5. ábra. $\alpha_i < 0$ eset

Miután a feltételezés szerint az első fázisban vagyunk, ezért $w(0)<0$ teljesül. Ha t értékét 0-ról t_1-re növeljük, $w(t)$ is növekszik, mégpedig

$$r_1 = -\left(\sum_{i \in M} \alpha_i - \sum_{i \in P} \alpha_i\right) \tag{5.13}$$

meredekséggel összesen $r_1 t_1 \geqq 0$ értékkel, mert r_1 éppen az 5.1. lemmában definiált d_j-nek a -1-szerese és két töréspont közt $w(t)$ lineáris. A $t=t_1$ pontban legalább egy bázisváltozó fizibilitási állapota megváltozik. Legyen i_1 annak a változónak az indexe, amely a t_1-et definiálja. Ha $\alpha_{i_1}<0$, akkor $f_{i_1}(t)$ a korlátját alulról érte el, így az i_1 index vagy az M halmazból átkerül F-be, vagy F-ből P-be. Ez utóbbi termé-

szetesen csak az $i_1 \in I_u \cap F$ esetben fordulhat elő. Mindkét esetben t_1-től kezdve $w(t)$ meredeksége $r_2 = r_1 + \alpha_{i_1}$ lesz, amint az (5.13)-ból kiolvasható. Minthogy $\alpha_{i_1} < 0$, ez azt jelenti, hogy $r_2 = r_1 - |\alpha_{i_1}|$. Ha $\alpha_{i_1} > 0$, akkor $f_{i_1}(t)$ felülről érte el a korlátját, így az i_1 index vagy P-ből átkerül F-be, vagy F-ből M-be. Mindkét esetben — amint az (5.13)-ból szintén leolvasható — $w(t)$ meredeksége t_1-től kezdve $r_2 = r_1 - \alpha_{i_1}$, amit $\alpha_{i_1} > 0$ miatt $r_2 = r_1 - |\alpha_{i_1}|$ alakban is fel lehet írni. Az adódott tehát, hogy t_1-től kezdve $w(t)$ meredeksége mindenképpen $|\alpha_{i_1}|$-gyel csökken. Miután ugyanez a gondolatmenet minden t_k $(k = 1, \ldots, Q)$ pontra elmondható, ezért igaz az, hogy egy tetszőleges töréspontban $w(t)$ meredeksége $|\alpha_{i_k}|$-val csökken, vagyis $r_{k+1} = r_k - |\alpha_{i_k}|$. Ezzel beláttuk a következő tételt.

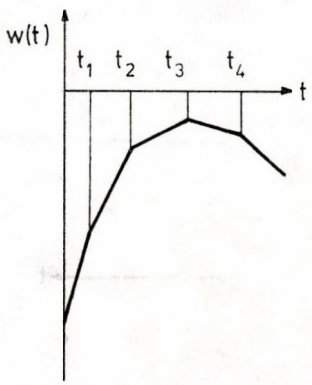
5.1. TÉTEL. $w(t)$ egy konkáv törtvonalfüggvény t_k $(k = 1, \ldots, Q)$ töréspontokkal, melynek $k+1$-edik lineáris intervalluma

$$r_{k+1} = r_k - |\alpha_{i_k}| \quad k = 1, \ldots Q$$

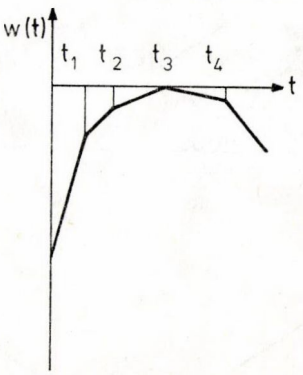
meredekségű. Egybeeső pontok esetén egy intervallum hossza 0 is lehet.

Megjegyzendő, hogy a bizonyítás során sehol sem használtuk ki azt, hogy a t_k értékek különbözők, így a tétel érvényessége minden speciális esetre is kiterjed.

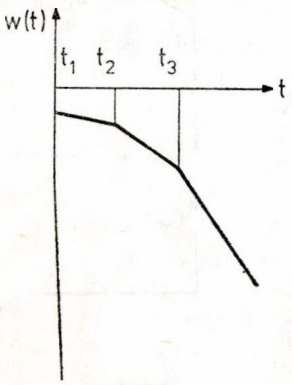
$w(t)$ néhány jellemző alakját a 6., 7. és 8. ábrán mutatjuk be.



6. ábra



7. ábra



8. ábra

A 6. ábra egy átlagos típusú alakot mutat. A 7. ábrán $w(t)$ eléri elméleti maximumát a 0-t. A 8. ábra érdekes jelenséget mutat. Most $t_1 = 0$, ami azt jelenti, hogy degenerált bázisnál vagyunk (mert vagy $t_1 = \beta_{i_1}/\alpha_{i_1} = 0$, vagy $t_1 = (\beta_{i_1} - u_{i_1})/\alpha_{i_1} = 0$) és az $|\alpha_{i_1}|$ érték r_1-hez képest olyan nagy, hogy $r_2 = r_1 - |\alpha_{i_1}| < 0$, tehát $w(t) < w(0)$ minden $t > 0$-ra. Ez azt jelenti, hogy a degeneráció és $|\alpha_{i_1}|$ nagysága megakadályozza azt, hogy $w(t)$ növekedjék. Ezt az esetet sokkal általánosabban fogjuk vizsgálni az 5.4. tétel tárgyalása során. A bemutatott alakok bővülhetnek azzal, hogy mindhárom esetben a maximumnál még vízszintes intervallumok lehetnek.

$w(t)$ maximalizálása során figyelembe kell vennünk az i) feltételt, vagyis, hogy a belépő változó fizibilis értéket vesz fel. Ha a bejövő x_j változó 1-es típusú, előfordulhat, hogy a $w(t)$-t maximalizáló t_k érték olyan nagy, hogy rá a $t_k \cong u_j$ teljesül. Ezt egy szerencsés speciális esetnek lehet tekinteni a szimplex iterációk szempontjából,

mert ekkor nem megyünk el $w(t)$ maximumáig, hanem megállunk $t=u_j$-nál és egy báziscsere nélküli iterációt végzünk oly módon, hogy az x_j változót az alsó korlátjáról átvisszük a felsőre. Ilyen iteráció során — melyet a továbbiakban 1-es típusú iterációnak nevezünk — új éta vektor nem keletkezik [27], ami számítástechnikailag igen kedvező, hiszen így nem növekszik az éta file behozásának munkaigénye és nem romlik a bázisinverzzel végzett műveletek pontossága.

Szem előtt tartva ezt az eshetőséget is, az eddigieket a következő tételben foglalhatjuk össze:

5.2. TÉTEL. Ha nem 1-es típusú iterációról van szó, akkor $w(t)$ egy $t=t_q$ pontban éri el globális maximumát, és ez olyan báziscserét határoz meg, ahol a kilépő i_q-adik bázisváltozó fizibilis értéken hagyja el a bázist és a belépő x_j változó is fizibilis értéket vesz fel. (Vagyis az így definiált báziscsere szimplex típusú iterációt határoz meg, melynek során azonban más bázisváltozók infizibilissé válhatnak.)

Bizonyítás. A tétel első része abból következik, hogy $w(t)$ egy szakaszonként lineáris konkáv függvény (5.1. tétel), míg a második rész azért igaz, mert minden t_k olyan báziscserét definiál, amihez a kilépő változónak fizibilis értéke tartozik (alsó, vagy felső korlát), miután minden t_k vagy egy T_{l_i}-vel vagy egy T_{u_i}-vel azonos, továbbá az i) feltétel teljesülése miatt a belépő változó is fizibilis.

Ha már (5.10) és (5.11) alapján meghatároztuk a T_{l_i} és T_{u_i} küszöbértékeket és az (5.12) szerinti sorba rendezést is elvégeztük, akkor könnyen megkaphatjuk $w(t)$ globális maximumát, mert ez egy olyan pontnál van, ahol $w(t)$ meredekségének előjele megváltozik. Ezt a következő lemmában fogalmazzuk meg:

5.3. LEMMA. Legyen $r_1=-d_j$ (>0) és számítsuk a következő rekurziót:

$$r_{k+1} = r_k - |\alpha_{i_k}| \quad k = 1, 2, \ldots . \tag{5.14}$$

Ekkor $w(t)$ maximumát az a q index határozza meg, melyre

$$\begin{aligned} r_q &> 0 \\ r_{q+1} &\leqq 0 \end{aligned} \tag{5.15}$$

teljesül.

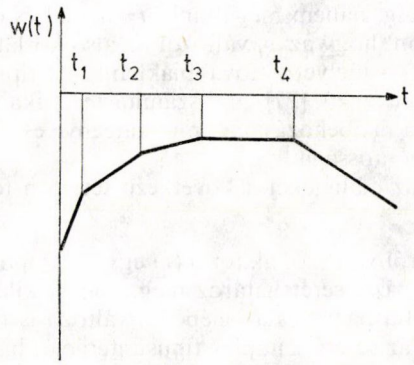
A továbbiakban q-t a maximumot szolgáltató t_k indexelésére fogjuk használni.

Megjegyezzük, hogy (5.15) helyett lehetett volna az

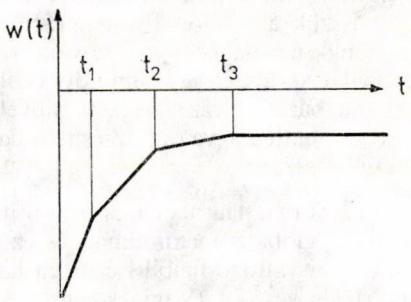
$$\begin{aligned} r_q &\geqq 0 \\ r_{q+1} &< 0 \end{aligned} \tag{5.15a}$$

kritériumot is használni. Ez bizonyos esetekben (5.15)-től különböző q értéket tud szolgáltatni. Ha ugyanis a maximumot szolgáltató helynek 1-nél nagyobb multiplicitása van, akkor előfordulhat, hogy ugyanazt a helyet (5.15a) eggyel nagyobb q indexszel adja meg, vagyis az (5.12) sorozatban két azonos értékű, de különböző indexű, és ilyen értelemben szomszédos pontra mutat a két kritérium. Előfordulhat azonban az is, hogy a maximum értéknél $w(t)$-nek vízszintes intervalluma van, mint a 9. és 10. ábrán.

A 9. ábrán látható esetben (5.15) a t_3 pontot, (5.15a) pedig a t_4 pontot fogja szolgáltatni azonos $w(t)$ érték mellett. A 10. ábrán viszont (5.15) a t_3-at adja, (5.15a)



9. ábra



10. ábra

azonban nem ad semmit, hiszen $w(t)$ meredeksége soha nem lesz negatív. Természetesen itt $Q=3$ miatt nincs is értelme továbbmenni, (5.15a)-t tehát külön ki kell egészíteni Q túllépésének a figyelésével, ami viszont nem szükséges (5.15)-nél.

Szükség van még $w(t)$ maximum értékének a meghatározására. Ennek egyik módja t_q-nak $w(t)$-be való behelyettesítése lehetne (5.9) szerint. Van azonban egyszerűbb lehetőség is, amely a következőképpen néz ki:

5.4. LEMMA. Legyen $t_0=0$ és számítsuk a következő rekurziót:

$$w(t_k) = w(t_{k-1})+(t_k-t_{k-1})r_k \quad k = 1, \ldots, q. \tag{5.16}$$

A q-adik lépésben éppen $w(t_q)$-t kapjuk.

A lemma helyességének belátásához elég az 5.1. tételre, valamint a t_k és r_k értékek definíciójára gondolni, illetve figyelembevenni a 6. ábrát.

Az eddig elmondottak képezik a $w(t)$ maximalizálásán alapuló kilépő vektor meghatározásnak, a DELPHI eljárásnak a lényegét. Ennek néhány további tulajdonsága előtt először a hagyományos pivotválasztás és a DELPHI kapcsolatát vizsgáljuk meg.

5.3. TÉTEL. A DELPHI eljárás annak a hagyományos pivotválasztási szabálynak az általánosítása, mely az első ütköző bázisváltozóig engedi növelni a belépő változót. Ha DELPHI-nél $k=1$-ig megyünk $w(t)$ maximalizálása helyett, akkor éppen ezt a hagyományos módszert kapjuk vissza.

Bizonyítás. A hagyományos szabálynál az (5.10) illetve (5.11) képletek által definiált nem-negatív küszöbértékek minimumát kell venni és ez lesz a belépő bázisváltozó nagysága, ugyanakkor a minimumot szolgáltató bázisváltozó kilép a bázisból. Ez viszont DELPHI-nél éppen annak felel meg, ha $w(t)$ maximalizálása helyett mindig csak a t_1 pontig mennénk el, t_1 (5.12)-beli definíciójának megfelelően, vagyis megállnánk $k=1$-nél. ($k=1$-nél teljesül az ii) feltétel.)

Ebből az is következik, hogy amennyiben az 5.2. tétel szerint, $w(t)$ maximalizálásával határozzuk meg a báziscserét, akkor minden lépésben legalább annyit haladunk előre, mint amennyit ugyanabban a helyzetben a hagyományos módszerrel mennénk. DELPHI konvergenciája a hagyományos módszer konvergenciájához

hasonlóan következik, ha degeneráció nem lép fel, ugyanis bázisismétlődés most épp úgy nem fordulhat elő, hiszen az infizibilitás mértékére nézve monoton bázisokon történnek az iterációk. A degeneráció esetével kicsit később foglalkozunk részletesen.

Az eddigiek során feltételeztük, hogy a bázisba belépő változó az alsó korlátjáról pozitív irányban mozdul el és az elmondottak erre az esetre voltak érvényesek. Ha a belépő változó negatív irányban lép be a bázisba (1-es típusú változó a felső korlátjáról jön be, vagy 3-as típusú negatív értékkel lép be), akkor az előző tárgyalásban α_i helyett $-\alpha_i$-t véve értelemszerűen igaz marad minden.

Miután a vázolt eljárástól nagyobb ugrások várhatók az előrehaladásban — DE*cent* *Leaps* in PH*ase* I — ezért kapta a DELPHI nevet.

Az előzőek alapján látható, hogy az infizibilitások száma egy-egy DELPHI iteráció során esetleg növekedhet, a mértéke azonban soha nem romolhat. Ugyanakkor az is igaz, hogy az infizibilitások száma iterációnként képes erősen csökkenni. Kedvező esetben a csökkenés száma közel q, vagy akár q is lehet (ha az (5.12)-nek megfelelő első q változó „jó" irányba mozdul el), ami nagy q érték esetén nagy javulást jelenthet.

A 0 típusú bázisváltozók esetén DELPHI előnyei általában nem tudnak érvényesülni. Az ugyan előfordulhat, hogy $w(t)$ maximum helyén egyszerre több 0 típusú bázisváltozó válik nullává, a bázisból azonban csak az egyikük fog kilépni. Amennyiben a megoldás ebben a pillanatban nem vált fizibilissé, akkor a bázisban bent maradt 0 típusúak ideiglenesen esetleg ismét infizibilissé válhatnak. Igaz, hogy az infizibilitás mértéke ez esetben viszont határozottan javulni fog. A 0 típusú bázisváltozókról lényegében tehát az mondható el, hogy azok általában még DELPHI-vel is csak egyesével tehetők fizibilissé.

Külön kell szólni a *degeneráció* esetéről, mely DELPHI számára kisebb valószínűséggel okoz problémát.

Egy bázis degenerált, ha legalább egy bázisváltozó a fizibilitási tartományának valamelyik határán (alsó vagy felső korláton) van. A degeneráció azért veszélyes, mert ilyenkor a belépő változó értékének infinitézimális növelése esetén az ilyen bázisváltozók — felső korláton levő változónál az $\alpha_i<0$ esetben, alsó korláton levő változónál az $\alpha_i>0$ esetben, 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.

DELPHI esetén a helyzet azonban másként alakulhat. Jelöljük h-val az (5.10), illetve (5.11)-gyel kiszámított 0 értékű küszöbértékek számát. Ez azt jelenti, hogy az (5.12) szerinti átrendezés után az

$$0 = t_1 = \ldots = t_h < t_{h+1} \leqq \ldots \leqq t_Q \tag{5.17}$$

relációt kapjuk. Az 5.3. lemma értelmében $w(t)$ a maximumát abban a t_q pontban veszi fel, melyre teljesül az, hogy

$$r_q > 0$$

és

$$r_{q+1} \leqq 0,$$

illetve részletesen kiírva

$$r_q = r_1 - \sum_{k=1}^{q-1} |\alpha_{i_k}| > 0$$

(5.18) és

$$r_{q+1} = r_1 - \sum_{k=1}^{q} |\alpha_{i_k}| \leqq 0.$$

Figyelembe véve, hogy $r_1 = -d_j$, ezért (5.18) így is írható

$$\sum_{k=1}^{q-1} |\alpha_{i_k}| < -d_j$$

(5.20)

$$\sum_{k=1}^{q} |\alpha_{i_k}| \geqq -d_j.$$

Ha az így definiált q-ra az teljesül, hogy $q>h$, akkor $t_q>0$, amint az (5.17)-ből látható, és ez esetben a degeneráció ellenére pozitív előrehaladás tehető az infizibilitás mértékében. Ezzel egyidejűleg az infizibilitások száma növekedhet. Most azonban ezt nem tekintjük kedvezőtlen jelenségnek, mert az így kijelölt báziscsere és a kapcsolódó transzformáció eredményeképpen az egyébként fizibilis 0 (illetve felsőkorlát) értékek helyettesítődnek fizibilis vagy infizibilis nem-nullákkal (illetve felsőkorláttól különböző értékekkel). Ezáltal az új bázis degeneráltsági foka csökken, ami növeli annak esélyét, hogy a soron következő iterációk minél kevésbé legyenek degeneráltak.

Ha $q \leqq h$, akkor $t_q=0$, vagyis a belépő változó 0 szinten kerül a bázisba. Ez azt jelenti, hogy a degeneráció megakadályoz egy pozitív előrehaladást $w(t)$-ben. Ennek a jelenségnek egy egyszerű esetét lehet látni a 7. ábrán.

A fent elmondottakat a következő tételben összegezzük:

5.4. TÉTEL. Ha

$$\sum_{k=1}^{h} |\alpha_{i_k}| < |d_j|, \tag{5.21}$$

ahol h értelmezése (5.17) szerint értendő, akkor az 5.2. tétel által meghatározott báziscsere végrehajtásával — a degeneráció ellenére — az infizibilitás $w(t)$ mértéke

$$D = w(t_q) - w(0) > 0 \tag{5.22}$$

értékkel javul. Ha (5.21) nem teljesül, akkor az előrehaladás nulla lesz.

Ezzel a DELPHI eljárás lényegének a bemutatása megtörtént. A továbbiakban még szólni kell arról, hogy DELPHI-nek milyen vonásai és milyen mértékben tekinthetők adaptívnak.

DELPHI használata esetén általában nagyon egyszerű lehetőség kínálkozik a *numerikus stabilitás* fokozására.

A szimplex módszer során a nem ritkán előforduló numerikus problémák gyakran fakadnak a pivot elem nem megfelelő nagyságából. A hagyományos pivotválasztás esetén ezt alig lehet elkerülni. Ha ugyanis a báziscserét definiáló minimális hányados egyértelműen adódik, akkor lényegében nincs lehetőség a pivot elem befolyásolására. Itt csupán azt lehet tenni, hogy ha az így adódó elem nem megfelelő, akkor az egész

oszlopot ideiglenesen ki lehet zárni a jelöltek közül. Ha a minimális hányados nem egyértelmű, akkor van lehetőség az egyenlők közül azt választani, melynek pivot eleme elfogadható nagyságrendű.

DELPHI használata esetén ilyen vonatkozásban sokkal nagyobb flexibilitás áll rendelkezésre. Ha a t_q-hoz tartozó pivot elem nagyságrendje nem megfelelő, akkor — $w(t)$ optimumát általában elveszítve — vissza lehet egyet lépni, ha $q>1$ volt és megnézni a $q:=q-1$-hez tartozó pivot elemet. Ha ez sem megfelelő, akkor ezt az eljárást folytatni lehet egészen $q=1$-ig. Másrészt az is igaz, hogy a maximumot adó q-t túl is lehet lépni a megfelelő méretű pivot elem keresése közben, $q:=q+1$-et véve mindaddig, amíg az így adódó q-ra $q\leqq Q$ és $w(t_q)\geqq w(t_0)$ egyszerre teljesül. Vannak esetek, amikor feltétlenül érdemes feladni $w(t)$ optimumát a megfelelő méretű pivot elem választása érdekében. Ily módon a DELPHI-vel működő LP algoritmusok megbízhatósága — esetenként jelentős mértékben — javulhat a konkrétan kialakult helyzet lehetőségeinek adaptív figyelembevételével.

Mindenképpen adaptívnak nevezhetjük DELPHI-nek azt a tulajdonságát, hogy képes észrevenni mennyire lehet egy adott bázis és belépő változó esetén az *infizibilitás mértékét maximálisan javítani*. Miután ezáltal az első fázisban — és mint később látni fogjuk összességében is — a szükséges iterációs lépések és a számítási munka csökkenését lehet elérni, így DELPHI alkalmazása a hatékonyság növekedését tudja eredményezni.

A numerikus stabilitásnál elmondottakkal legalább egyenértékűen fontos adaptív tulajdonsága DELPHI-nek az, hogy — akár erősen — *degenerált bázisok esetén is hatékony lépések*et tud tenni, amint az az 5.4. tételnél szerepelt. Ez nemcsak az egész eljárás hatékonyságát növeli, hanem a megbízhatóságát is, mert most jelentősen kisebb a degenerációból származó „üresjáratokra", illetve ciklizálásra való hajlam. Itt pusztán a történelmi hűség kedvéért meg kell említeni, hogy a DELPHI-vel és a később tárgyalandó ADACOMP-pal kapcsolatos gondolatok éppen az erősen degenerált LP feladatok megoldása (illetve meg nem oldása) során szerzett tapasztalatok alapján születtek.

5.1.2. *DELPHI implementálása*

DELPHI implementálására és használatára vonatkozólag meg kell még vizsgálni néhány kérdést. DELPHI *memóriaigénye* elvileg nagyobb mint a hagyományos módszeré, mert tárolni kell az (5.10) és (5.11) alatt definiált küszöbértékeket. Ezek maximális száma $2(m-1)$, mert a célfüggvénysor kivételével minden feltételre legfeljebb két küszöbérték definiálódik. Tárolás után ezeket az értékeket (5.12) szerint rendezni kell. Ez történhet „in situ", így újabb helyigény itt nem lép fel, viszont tárolni kell egy Q elemű permutáció vektort az (5.12)-beli t_k-k és az eredeti (5.10) és (5.11) értékek közötti megfeleltetés feljegyzésére. Tekintettel arra, hogy — amint az a 4.2. pontban szerepelt — DELPHI során sem a szimplex szorzóra, sem az éta vektorokra nincs szükség, azok helye használható a t_k-k és a permutáció vektor tárolására, így DELPHI-nek csak akkor van többlet memóriaigénye, ha ezek a területek nem elegendőek.

A DELPHI-vel kapcsolatos többlet *számítási munka* meghatározásánál először is azt kell figyelembe venni, hogy a T_{l_i} és T_{u_i} küszöbértékek meghatározása alig

jelent valami többlet munkát, mert ezen mennyiségek általában nagy részét a hagyományos módszer is kiszámítja. Ezután végre kell hajtani az (5.12) szerinti rendezést. Ha Q nagy, akkor az összes Q elem rendezése költséges lehet. Erre azonban általában nincs szükség. Egy egyszerű tapasztalat felhasználásával elkerülhető a „fölösleges" rendezés. Arról van szó, hogy a szimplex eljárás kezdeti szakaszán Q értéke m-hez képest többnyire kicsinek adódik, és ilyenkor a $w(t)$ maximumát szolgáltató q index $Q/2$ körül ingadozik. Az iterációk előrehaladtával Q értéke hajlamos a növekedésre, q azonban 8 (gyakran 3) alá esik. Ebben a helyzetben fölösleges a Q elem teljes rendezése, hiszen csak a néhány legkisebbre van szükség. Ezért az a célszerű, ha olyan rendező eljárást használunk, amelyik alulról fokozatosan építi fel a rendezettséget és a k-adik lépés után az első k elem helyes sorrendbe kerül. Így a rendezés lépéseivel párhuzamosan tudjuk számolni az (5.14) és (5.16) rekurziókat és amikor az (5.15) megállási feltétel teljesül, a rendezéssel is leállunk. Így a t_k $(k=q+1, \ldots, Q)$ elemek rendezésére nem kerül sor. A Q és q-ra vonatkozó tapasztalatok tükrében ilyen módon jelentős megtakarítás érhető el olyan rendezési eljárásokkal szemben, melyek csak a legvégén szolgáltatnak egy helyesen rendezett számsort.

DELPHI implementálásával kapcsolatban felmerül az a kérdés is, hogy hogyan illeszthető be a többszörös kiválasztás és az ezt követő *szuboptimalizálás* kereteibe. A 2.3. pontban tárgyaltuk, hogy a többszörös kiválasztás során a mátrix egyszeri átnézése alkalmával több javító vektor jelöltet választunk ki, és ezek közül további szempontok alapján határozzuk meg a bázisba ténylegesen beléptetendő oszlopot. Ezek a további szempontok bizonyos hierarchiát alkotnak, melyek — 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 a DELPHI technika alkalmazása esetén hogyan érvényesíthető ez az elv.

Tegyük fel, hogy a kiválasztott oszlopok száma N_s. A hagyományos módszer először meghatározza minden oszlopra a pivot elemet a hozzá tartozó $\theta_j (j=1, \ldots, N_s)$ hányadossal együtt [27], ami egyébként a belépő változó elmozdulásának nagyságát jelenti, és amelyik oszlopra a $|d_j \theta_j|$ szorzat, vagyis a tényleges előrehaladás a legnagyobb, azt vonja be a bázisba. DELPHI használata esetén a tényleges előrehaladást nem egy $d_j \theta_j$ jellegű lineáris kapcsolat határozza meg, hanem a j-edik kiválasztott oszlophoz tartozó $w_j(t)$ nemlineáris függvény maximumának és kezdeti értékének különbsége, D_j, amint az (5.22)-ből, a j index értelemszerű beiktatásával 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 (5.16) 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 DELPHI különösen az erősen degenerált bázisok esetén tud igen hatásos lenni. Az 5.4. tétel megmutatta, hogy egyetlen oszlop esetén milyen esély van a degenerált báziscsere elkerülésére. Ez a lehetőség most megsokszorozódik és ha a jelölt oszlopok közt csak egy is $D_j>0$ értéket ad, akkor biztos, hogy nem degenerált báziscsere történik és ezzel együtt várhatóan a bázis degeneráltsági foka is csökken, ami tovább javítja az esélyt a későbbi hatékony iterációs lépésekre.

Az eddig elmondottak már eléggé rávilágítottak DELPHI-nek egy nem elhanya-

golható előnyére, nevezetesen arra, hogy rendkívül könnyen beépíthető tetszőleges — a primál szimplex módszeren alapuló — LP rendszerbe. Ennek az igazságáról személyesen is meggyőződtem, amikor DELPHI-t beépítettem a korábban már említett LIPROS LP programcsomagba. A beépítés oly módon történt, hogy a régi PIVOT1 szubrutint helyettesítettem DELPHI-vel. (Ez DELPHI-re nézve azzal a hátránnyal járt, hogy kénytelen volt azokkal a kiválasztott oszlopokkal dolgozni, melyeket a PIVOT1 számára kidolgozott eljárás határozott meg. Noha DELPHI már így is nagyon kedvezőnek bizonyult, célszerűnek látszott új oszlopkiválasztási módszert kidolgozni. Ez vezetett a később tárgyalandó ADACOMP eljárás elkészítéséhez.) Tekintettel arra, hogy a számítógépek központi egysége 2—3 nagyságrenddel gyorsabb (4.1. pont), mint a háttértárral való kommunikáció, ezért általában azok az eljárások az előnyösek, melyek főleg a központi memóriában dolgoznak és csak kevésszer fordulnak a háttértárhoz. DELPHI-vel is ez a helyzet, mert ez egyáltalán nem igényel külön I/O műveletet, sőt segítségével éppen sok I/O-t lehet megtakarítani azáltal, hogy alkalmazása az össz iterációszámot, így a szükséges major iterációk (2.3. pont: RSM2 eljárás) számát is jelentősen csökkenteni tudja.

A kísérleti összehasonlító futások során azt tapasztaltam, hogy PIVOT1 és DELPHI vonatkozásában az egy iterációra eső idő gyakorlatilag változatlan (±5%-os határon belül) volt. A következő táblázaton több, mint száz futásból kiválasztott néhány tipikusnak talált futás eredményét mutatom be. A futások teljesen azonos körülmények (futási paraméterek, memóriaméret, stb.) között zajlottak le. Az induló bázis minden esetben a tisztán logikai változókból álló triviális bázis volt. A táblázat az első fázisbeli lépések számát mutatja, a feladat méreténél a sorok (m) és strukturális oszlopok száma (n) szerepel, DOD az induló bázis degeneráltsági fokát (a bázismegoldás nulla elemeinek aránya m-hez) jelenti %-ban.

Néhány további — a megoldás menetére vonatkozó — megjegyzést kell még tenni.

A 2. feladat csupa $\geqq$ típusú feltételből állt (tehát nem volt benne egyelőség, így 0 típusú változó sem!). Az infizibilitások száma két esetben kicsit növekedett, más esetekben viszont „drasztikus" módon csökkent (1-1 lépésben 10—20 infizibilitás is megszűnt). Tekintettel arra, hogy most 0 típusú változó egyáltalán nem volt a bázisban, DELPHI potenciális előnyei jól tudtak érvényesülni (lásd 5.3. tétel utáni megjegyzéseket!).

A 4. feladatot PIVOT1-gyel nem sikerült megoldani. Az eljárás az iterációk hosszú során nem tudott elmozdulni az erősen degenerált állapotról és — bár ciklus-

Feladat		PIVOT1	DELPHI	Megjegyzés
sorszám	méret			
1.	62×70	43	34	25 egyenlőség a feltételek között
2.	61×10	42	12	Az induló bázis minden változója infizibilis
3.	100×130	97	74	50 egyenlőség a feltételek között
4.	170×120	megoldhatatlan	278	60 felső korlátos változó, DOD=96%
5.	237×184	—	107	55 felső korlátos változó, DOD=99,5%

ba nem esett — a numerikus hibák kedvezőtlen halmozódása miatt (4 byte-os lebegőpontos aritmetika használata mellett) az adott keretek közt a feladatot nem tudta megoldani, amint ezt az ötszázadik iteráció körül kijelezte. DELPHI is sokat „birkózott" a degenerált bázisokkal, de a 205. iterációban talált egy pozitív előrehaladást, mégpedig annak 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 (DELPHI nagyobb numerikus stabilitást tud biztosítani), minden közbülső újrainvertálás jól reprodukálta az invertálás előtti állapotot és a végeredmény is pontosnak adódott.

Az 5. feladat csak DELPHI-vel futott le, így ez valójában nem összehasonlító futás. A megoldás menete érdekes és egyben jellemző tapasztalatokat hozott. Az induló triviális bázisra alkalmazott CRASH menet után keletkezett új bázis is igen erősen degenerált volt. DELPHI az infizibilitások számát és mértékét néhány lépésben jelentősen lecsökkentette. Ezután egy hosszabb „üresjárat" következett: 60 iteráción keresztül csak degenerált lépések történtek, minden előrehaladás nélkül. A 90. iteráció során az infizibilitások száma 1-ről 97-re szökött fel, ugyanakkor $w(t)$ egy pozitív lépést tett előre. A megoldásbeli nulla értékek zöme „elromlott", ami után nagyon hatékony iterációs lépések történtek egészen a 107. lépésig, amikor is az első fázis befejeződött. A DELPHI működésére vonatkozó néhány más jellegű statisztikai adat a Függelék I. pontjában található.

5.1.3. *DELPHI egy általánosítása*

Ha relaxáljuk az i) feltételt is, vagyis megengedjük, hogy a bázisba belépő változó is infizibilis értéket vegyen fel, akkor egy még általánosabb esethez jutunk.

A bázisba belépő változó kétféleképpen lehet infizibilis:

— ha felső korlátjánál nagyobb értékkel lép be a bázisba, vagy
— ha negatív értékkel lép be a bázisba.

Nyilvánvaló, hogy ha a belépő változó 3-as típusú, akkor az mindenképpen fizibilis értéket vesz fel, így a további tárgyalásból kizárjuk. Tegyük fel, hogy egy x_j bázison kívüli változó 0 szinten van és ezt akarjuk behozni a bázisba valamilyen szinten. Jelöljük $W_j(t)$-vel a hozzátartozó infizibilitási függvényt. Könnyen látható, hogy $W_j(t)$, vagyis az infizibilitások mértéke, ha az x_j változó értékét t-re változtatjuk, az alábbi módon néz ki:

$$W_j(t) = \sum_{i \in I} (\beta_i - t\alpha_{ij})^- - \sum_{i \in I_u} (\beta_i - t\alpha_{ij} - u_i)^+ + G_j(t), \tag{5.23}$$

ahol

$$G_j(t) = \begin{cases} t, & \text{ha} \quad t < 0 \\ u_j - t, & \text{ha} \quad t > u_j \\ 0, & \text{különben.} \end{cases} \tag{5.24}$$

Felhasználva (5.9)-et (5.23) a következőképpen is felírható

$$W_j(t) = w(t) + G_j(t). \tag{5.23a}$$

Először azt vizsgáljuk meg, hogy mi a szükséges feltétele annak, hogy egy nem 0-típusú x_j valamilyen irányú elmozdítása javítani tudjon az infizibilitás mértékén. Erről szól a következő lemma.

5.5. LEMMA. Ahhoz, hogy egy 0 szinten levő bázison kívüli változó elmozdítása javítson az infizibilitás mértékén, az szükséges, hogy

I. pozitív irányú elmozdulás esetén

$$d_j = \sum_{i \in M} \alpha_{ij} - \sum_{i \in P} \alpha_{ij} < 0. \tag{5.25}$$

II. negatív irányú elmozdulás esetén

$$d_j = \sum_{i \in M} \alpha_{ij} - \sum_{i \in P} \alpha_{ij} > 1 \tag{5.26}$$

teljesüljön.

Bizonyítás. Ha x_j pozitív irányban mozdul el, akkor, elég kicsi $t>0$ esetén $G_j(t)=0$, az (5.24)-beli definícióból következően. Ekkor viszont az 5.1. lemma feltételei teljesülnek, így (5.25) szükségessége az 5.1. lemmából következik. Ha x_j negatív irányban mozdul el, akkor — ugyancsak az 5.1. lemma felhasználásával — $W_j(t)$ megváltozása:

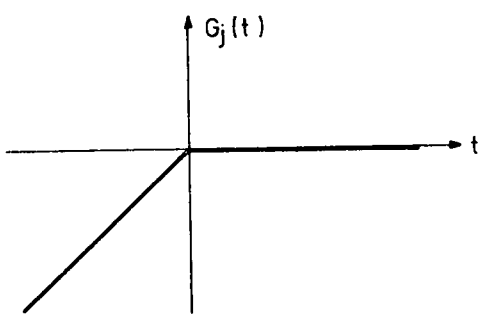
$$\Delta W_j = W_j(t) - W_j(0) = -td_j + \Delta G_j = -td_j + G_j(t), \tag{5.27}$$

mert $G_j(0)=0$. Tekintettel arra, hogy $t<0$-ra $G_j(t)=t$, ezért (5.27) a

$$\Delta W_j = -td_j + t \tag{5.28}$$

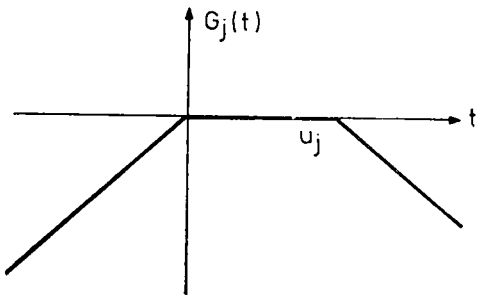
alakban írható. Itt $\Delta W_j>0$-hoz $d_j>1$ szükséges.

Amennyiben x_j 2-típusú, úgy $G_j(t)$ a következőképpen néz ki:



11. ábra

Ha x_j 1-típusú, úgy $G_j(t)$ alakja:



12. ábra

$G_j(t)$ tehát mindkét esetben konkáv függvény.

Ha x_j-t elmozdítjuk 0-ról, akkor a bázisváltozók értéke (5.5)-nek megfelelően megváltozik. Eközben a fizibilitási állapotuk is megváltozhat, ha átlépik fizibilitási tartományuk valamelyik határát. Ezek a küszöbértékek (5.7) alapján határozhatók meg. Külön vizsgáljuk a $t \geqq 0$ illetve a $t \leqq 0$ esetet.

I. Ha $t \geqq 0$, vagyis x_j változót pozitív irányban mozdítjuk el, akkor a keresett pontokat az (5.10) és (5.11) alatt definiált küszöbértékek adják, melyekhez — amennyiben x_j 1-típusú — egy további pont jön hozzá, nevezetesen az, ahol x_j átlépi felső korlátját. Ezt a

$$T_u = u_j/1 \tag{5.29}$$

küszöbértékként tartjuk nyilván, vagyis egy olyan „tört"-ként, melynek nevezője 1. Ily módon ez a töréspont egyenrangúvá vált a DELPHI-nél definiált töréspontokkal. A hozzátartozó α_i érték 1-gyel egyenlő.

II. Ha $t \leqq 0$, vagyis x_j változót negatív irányban mozdítjuk el, akkor a keresett töréspontokat az alábbi módon kapjuk meg:

$$\left.\begin{array}{l} T_{l_i} = \beta_i/\alpha_i < 0 \quad (\alpha_i \neq 0) \\ \text{vagy} \quad T_{l_i} = 0, \quad \text{ha} \quad \beta_i = 0 \quad \text{és} \quad \alpha_i < 0 \end{array}\right\} i \in I \tag{5.30}$$

$$\left.\begin{array}{l} T_{u_i} = (\beta_i - u_i)/\alpha_i < 0 \quad (\alpha_i \neq 0) \\ \text{vagy} \quad T_{u_i} = 0, \quad \text{ha} \quad \beta_i = u_i \quad \text{és} \quad \alpha_i > 0 \end{array}\right\} i \in I_u. \tag{5.31}$$

Ezekhez jön hozzá mindenképpen egy $T_u = 0$ érték a belépő változó saját alsó korlátjának átlépése miatt, amihez egy $\alpha = 1$ érték tartozik.

Ezen kifejezések értelmezésénél az (5.10) és (5.11)-nél elmondott megjegyzések érvényesek.

Az I., illetve II. esetnek megfelelő töréspontokat (5.12)-höz hasonlóan nagyság szerint sorba rendezzük:

$$0 \leqq t_1 \leqq \ldots \leqq t_{Q_1}, \tag{5.32a}$$

illetve

(5.32b) $$t_{Q_2} \leqq \ldots \leqq t_1 \leqq 0.$$

Ezek után az 5.1. tétel analogonja a következőképpen néz ki:

5.5. TÉTEL. $W_j(t)$ egy konkáv törtvonalfüggvény az (5.32a) illetve (5.32b) szerinti t_k töréspontokkal. Ha az $r_1 = -d_j$ jelölést használjuk, akkor $W_j(t)$ $k+1$-edik lineáris intervalluma az I. esetben

(5.33) $$r_{k+1} = r_k - |\alpha_{i_k}| \quad k = 1, \ldots, Q_1,$$

a II. esetben pedig

(5.34) $$r_{k+1} = r_k + |\alpha_{i_k}| \quad k = 1, \ldots, Q_2$$

lesz.

A tétel bizonyítása értelemszerűen következik az 5.1. tételnél elmondottakból. $W_j(t)$ alakja az I. esetben olyan lesz mint amilyet a 6., 7., 8. ábrán bemutattunk, II. esetben pedig annak a függőleges tengelyre vett tükörképe, azzal a különbséggel, hogy ilyenkor mindig $t_1 = 0$ lesz.

Ugyancsak értelemszerűen lehet használni az 5.3. lemmát $W_j(t)$ maximum helyének és 5.4. lemmát $W_j(t)$ maximum értékének a meghatározására.

Az 5.2. tétel igazsága most az 1-es típusú iterációkra tett kitétel nélkül áll fenn, hiszen a belépő változó túl léphet felső korlátján. Azt azonban célszerű megnézni, hogy az 5.3. lemma szerinti maximum keresés milyen esetben vezet túl a $t_k = u_j$ ponton. Ehhez az (5.15) megállási kritériumot kell figyelembe venni. A t_k ponthoz a k-adik intervallum vezet, ennek meredeksége r_k. A t_k pont utáni intervallum meredeksége

$$r_{k+1} = r_k - 1$$

lesz (5.29) miatt. Ha $r_{k+1} > 0$, akkor a megállási kritérium még nem teljesül, így t_k-n tovább kell menni. $r_{k+1} = r_k - 1 > 0$ viszont akkor áll fenn, ha

(5.35) $$r_k > 1.$$

Ha tehát $W_j(t)$ meredeksége $t = u_j$-ig nem csökkent le 1 alá, akkor $W_j(t)$ maximuma olyan helyen adódik, ahol a belépő változó már infizibilis értékű lesz.

Érdemes felfigyelni arra, hogy mennyire analógok az (5.26) és (5.35) kritériumok, vagyis annak a feltételei, hogy a belépő változó infizibilis értéke javítani tudjon az infizibilitások mértékén.

A $W_j(t)$ maximalizálásán alapuló első fázis eljárás DELPHI általánosítását jelenti és a továbbiakban DELPHI-A néven fogunk hivatkozni rá.

DELPHI-A implementációja jelen sorok írásakor még nem készült el, így számítástechnikai tapasztalatokról nem tudok beszámolni. Esetleges hasznosságát csak az eddigi egyébirányú tapasztalatok alapján lehet megbecsülni. Az első fázis elején valószínűnek látszik, hogy előfordulhat olyan eset, amikor a belépő változó infizibilis értéke mellett lehet az infizibilitás mértékét a legjobban javítani. Ilyenkor ugyanis a szereplő árnyékárak amplitúdói abszolút értékben elég nagyok, így jó esély van arra, hogy vagy az (5.26) vagy az (5.35) feltétel teljesüljön. Az első fázis előrehaladtával az amplitúdók általában csökkennek, aminek következtében ez az esély is kisebb lesz. Mindenesetre az látszik célszerűnek, hogy — ha ez nem adódik

automatikusan az árnyékárak amplitúdójából — az első fázis vége felé a belépő változó inkább fizibilis legyen, ellenkező esetben ugyanis az újabb infizibilitások eliminálására — monoton $W_j(t)$ mellett is — további „kis" iterációkra lehet szükség. Másszóval, ilyen esetben DELPHI-A használata helyett inkább DELPHI ígérkezik jobbnak.

5.2. *MPRICE*

A 2. fejezetben RSM1 illetve RSM2 tárgyalásánál nyitva hagytuk azt a kérdést, hogy az elvileg jó, potenciális javítóvektorok közül melyiket léptetjük be a bázisba, illetve melyeket választjuk ki a szuboptimalizálás számára. Ebben a pontban ezen kérdés néhány részletéről lesz szó.

Régóta ismeretes, hogy az oszlopkiválasztás milyen jelentősen befolyásolja az LP feladatok megoldásához szükséges iterációk számát illetve számítási össz munkát. A legegyszerűbb kritériumok (első jó oszlop, legnegatívabb d_j) „olcsón" valósíthatók meg, viszont általában elég kevéssé hatékonyak a bonyolultabb és munkaigényesebb kritériumokhoz képest. Tapasztalat azt mutatja, hogy a teljes mátrixra vonatkozó legnagyobb előrehaladás elvének alkalmazása esetén van szükség a legkevesebb iterációra, ennek a megvalósítása viszont igényli a teljes transzformált tablót minden iterációs lépésben, ami még kisebb méretű feladatok esetén is gyakorlatilag megvalósíthatatlan a nagy helyigény, a sok számítási és háttérkommunikációs munka miatt, amint arról a 2.2. pontban szó volt.

A módosított szimplex módszer alapján működő LP rendszereknél a d_j korrigált árnyékár az MPRICE eljárás során a (2.11)-ben szereplő $d_j = \boldsymbol{\pi}^T \mathbf{a}_j$ skalár szorzattal áll elő. Ebből azonban még semmit nem lehet tudni arra nézve, hogy x_j mekkora értéken tud belépni a bázisba és így végül is mekkora lesz a célfüggvény szerinti előrehaladás. Erre vonatkozóan többen folytattak vizsgálatokat és születtek viszonylag jól becslő de drágán, illetve nehezen implementálható módszerek ([4] illetve [10]), ugyanakkor készült a gyakorlatban jól bevált, „pontatlanabb", de hatékonyan implementálható eljárás is [3]. Ezek általában az x_j várható nagyságára vonatkozó közelítéseket számítanak ki és ez, valamint d_j alapján becsülik a célfüggvény várható javulását. LIPROS is fel van szerelve egy ilyen technikával [23]. Valamennyi eljárás adaptív módon működik és az aktuálisan kialakult helyzet függvényében határozza meg azokat a dinamikus normáló faktorokat, melyekkel a megfelelő d_j-ket elosztva, — az eredeti normálástól függetlenül — egymással összehasonlítható árnyékárak adódnak abban az értelemben, hogy a nagyobb amplitúdójú árnyékárhoz várhatóan nagyobb előrehaladás tartozik. Mindez, még a legegyszerűbb esetben [23] is, egy m méretű segédvektor BTRAN-ját és a jó előjelű árnyékárral rendelkező oszlopokra nézve egy többlet skalár szorzást jelent major iterációnként. Ennek további részleteiről itt nem kívánunk szólni.

A figyelmet most két dologra fordítjuk. Először azt nézzük meg, hogyan lehet olyan oszlopokat kiválasztani, melyeken nagyobb valószínűséggel lehet nem degenerált, tehát pozitív előrehaladást eredményező lépéseket tenni. Ezután egy olyan adaptív oszlopkiválasztási eljárást vezetünk be, mely lehetőséget biztosít arra, hogy az első fázisban a valódi célfüggvény szempontjai is minél jobban érvényesülni tudjanak, így az első fizibilis megoldás lehetőleg minél közelebb legyen az optimális megoldáshoz.

5.2.1. *Egy degeneráció elleni stratégia*

Jelöljük H-val azon bázisváltozók indexhalmazát, melyek a fizibilitási tartományuk valamelyik korlátján vannak. Ezen belül legyen

$$H_1 = \{i : \beta_i = 0\} \tag{5.36}$$

és

$$H_2 = \{i : \beta_i = u_i, \text{ ha } u_i \text{ definiálva van}\}. \tag{5.37}$$

Nyilvánvalóan $H = H_1 \cup H_2$. Egy bázis akkor degenerált, ha H nem üres halmaz.

5.6. LEMMA. Legyen $\mathbf{a}_j$ egy potenciális javító vektor. Jelöljük — az eddigieknek megfelelően — $\boldsymbol{\alpha}_j$-vel ennek az oszlopnak az aktuális bázisbeli képét ($\boldsymbol{\alpha}_j = \mathbf{B}^{-1}\mathbf{a}_j$). A j-edik oszlop bevonása pozitív előrehaladást eredményez, ha

$$\sum_{i \in H} |\alpha_{ij}| = 0 \tag{5.38}$$

teljesül.

Ha $H = \emptyset$, akkor a szumma értékét definíciószerűen 0-nak tekintjük.

Bizonyítás. Az (5.38) feltétel azt jelenti, hogy $\alpha_{ij} = 0$, ha $i \in H$. A kilépő változó meghatározására szolgáló hányadosokat viszont csak olyan sorokra kell képezni, melyekre $\alpha_{ij} \neq 0$ ((5.10) és 5.11)). H definíciója miatt ebben az esetben viszont csak nullától különböző értékű hányadosok értelmeződnek, így ezek külön-külön is, tehát minimumuk is (ez a második fázisban érdekes) nullától különböző. Ez viszont éppen a belépő változó elmozdulásának nagyságát jelenti.

Megjegyzés. DELPHI esetén a pozitív előrehaladáshoz az is elég, hogy $\sum_{i \in H} |\alpha_{ij}| < |d_j|$ teljesüljön, amint ez az 5.4 tételből látható.

Az 5.6. lemma természetesen csak egy elégséges feltételt mond ki. Az (5.10) és (5.11) küszöbértékek alaposabb vizsgálata azt is elárulja, hogy a most elmondottaknál kevesebb is elég. Ha ugyanis egy határon levő bázisváltozó mellett olyan előjelű α_{ij} érték áll, hogy a bázisváltozó jó irányba, vagyis fizibilitási tartományának belseje felé mozdul el, akkor ez nyilván nem akadályozza meg a pozitív előrehaladást. (Egy 0 szinten levő 0 típusú bázisváltozó indexe a H_1-hez is és a H_2-höz is hozzátartozik.) Felhasználva az (5.8)-ban bevezetett jelölést, a fentiek a következőt jelentik:

5.7. LEMMA. Egy $\mathbf{a}_j$ potenciális javítóvektor bevonása pozitív előrehaladást eredményez, ha

$$\sum_{i \in H_1} \alpha_{ij}^- + \sum_{i \in H_2} \alpha_{ij}^+ = 0 \tag{5.39}$$

teljesül.

Látható, hogy ha egy 0 típusú változó 0 szinten van, akkor (5.39) teljesüléséhez az kell, hogy a megfelelő $\alpha_{ij} = 0$ legyen.

Mindebből az is következik, hogy a második fázisban egy 0 típusú bázisváltozó, mely mellett $\alpha_{ij} \neq 0$ áll, megakadályozza a pozitív előrehaladást. Az első fázisban DELPHI, vagy DELPHI-A alkalmazása esetén ez természetesen nem feltétlenül igaz, amint az az 5.4. tételből könnyen kiolvasható.

Fentiek alapján tehát degenerált bázis esetén az lenne célszerű, ha a szuboptimalizálás céljaira olyan vektorokat tudnánk kiválasztani, melyekre az 5.6. vagy az 5.7. lemma feltételei teljesülnek. A módosított szimplex módszer alkalmazása esetén azonban az α_{ij} együtthatók nem állnak rendelkezésre, így a feltételek ebben a formában nem ellenőrizhetők. Ehelyett azonban lehet próbálkozni az ott definiált szummák közelítő meghatározásával az alábbi módon.

Az (5.38) szumma közelítésére az

$$\left|\sum_{i\in H}\alpha_{ij}\right| \tag{5.40}$$

szummát tekintjük. Ez ugyanis a következő egyszerű módon számítható. Legyen $\mathbf{c}$ egy olyan m dimenziós vektor, melynek i-edik komponense:

$$c_i=\begin{cases}1, & \text{ha } i\in H\\ 0, & \text{különben.}\end{cases} \tag{5.41}$$

Ezzel a $\mathbf{c}$ vektorral

$$\left|\sum_{i\in H}\alpha_{ij}\right| = |\mathbf{c}^T\boldsymbol{\alpha}_j| = |\mathbf{c}^T\mathbf{B}^{-1}\mathbf{a}_j|. \tag{5.42}$$

Tekintettel arra, hogy ebben a $\mathbf{c}^T\mathbf{B}^{-1}$ vektor j-től független, ezért egy BTRAN művelettel meghatározható. Ezután a potenciális javító vektorok közül azokat részesíthetjük előnyben, melyekre az újabb skalárszorzatként (5.42)-ből adódó mennyiség közelebb lesz a nullához. Az (5.38) kifejezés (5.40)-nel történő helyettesítése természetesen esetenként nagy eltérést is okozhat, mert $\left|\sum_{i\in H}\alpha_{ij}\right|=0$-ból egyáltalán nem következik a $\sum_{i\in H}|\alpha_{ij}|$ nulla volta, mert csak az igaz, hogy $\left|\sum_{i\in H}\alpha_{ij}\right|\leqq\sum_{i\in H}|\alpha_{ij}|$.

Tekintettel azonban arra, hogy az (5.40) szumma számítása (5.42) alapján igen „olcsó" művelet még a szükséges BTRAN-nal együtt is (ami párhuzamosan mehet más vektorok BTRAN-jával), és a gyakorlatban sokszor jól jelzi a különbséget a potenciális javító vektorok közt, ezért érdemes használni a nem degenerált iterációkat elősegítő vektorok keresésére. Meg kell jegyezni, hogy ilyen jellegű közelítések nem vezetnek az algoritmus helytelen működéséhez, mert csak a matematikailag egyformán jó vektorok köréből történő válogatásra használjuk fel a közelítő információt. Becslések használata egyébként sem idegen az LP implementációk gyakorlatában [23], [3], [10].

Az 5.7. lemma feltételével kapcsolatban pontosabb állítást lehet megfogalmazni — bár sajnos csak negatív értelemben.

5.8. LEMMA. Ha

$$\sum_{i\in H_1}\alpha_{ij}-\sum_{i\in H_2}\alpha_{ij}>0, \tag{5.43}$$

akkor a második fázisban csak degenerált lépés tehető.

Bizonyítás: Az (5.43) szumma csak úgy lehet pozitív, ha vagy legalább egy $i\in H_1$-re $\alpha_{ij}>0$, vagy legalább egy $i\in H_2$-re $\alpha_{ij}<0$. Ez viszont éppen azt jelenti, hogy egy korláton levő bázisváltozó a fizibilitási tartományából kifelé akar elmozdulni, így a bázis fizibilitásának megtartása csak úgy lehetséges, ha nulla szintű báziscsere történik.

Az (5.43) feltétel teljesülése könnyen ellenőrizhető. Legyen ugyanis most $\bar{\mathbf{c}}$ egy olyan m dimenziós vektor, melynek i-edik komponense:

$$\bar{c}_i = \begin{cases} 1, & \text{ha } i \in H_1 \\ -1, & \text{ha } i \in H_2 \\ 0, & \text{különben.} \end{cases} \tag{5.44}$$

Ezzel az (5.43) szumma a

$$\bar{\mathbf{c}}^T \boldsymbol{\alpha}_j$$

alakban írható fel. Ez viszont tovább alakítva

$$\bar{\mathbf{c}}^T \boldsymbol{\alpha}_j = \bar{\mathbf{c}}^T \mathbf{B}^{-1} \mathbf{a}_j \tag{5.45}$$

lesz. Itt a $\bar{\mathbf{c}}^T \mathbf{B}^{-1}$ vektor már független j-től és meghatározására nézve az (5.42)-nél elmondottak teljes egészében érvényesek. A felhasználására vonatkozóan az mondható el, hogy segítségével — amíg van több javító vektor jelölt — ki lehet szűrni olyan oszlopokat, melyeken a második fázisban biztos, hogy csak degenerált lépést lehet csinálni.

A fentiekben leírt két módszer — noha garanciát nem nyújt, de — hasznos és viszonylag olcsó segédeszközt jelent mind első, mind második fázisban ahhoz, hogy degenerált bázisok esetén növekedjék a nem degenerált báziscserét lehetővé tevő oszlopok kiválasztása.

5.2.2. *Az ADACOMP eljárás*

A degeneráción kívül a szimplex módszernek van egy másik kritikus pontja: az első fázis. Az első fizibilis megoldás keresése közben ugyanis az igazi célfüggvény szempontjai általában figyelmen kívül maradnak, ami gyakran azt eredményezi, hogy sok iterációra van szükség a második fázisban. Minthogy DELPHI-nél az a látszat alakulhat ki, hogy túlságosan „hajszolja" a fizibilitás elérését, ezért felmerülhet a gyanú, hogy eközben nem távolodunk-e el túlságosan az LP optimumától. (Ez a kérdés természetesen nem merül fel, ha csak a feltételrendszer konzisztenciáját kell eldönteni.) Válaszként röviden azt lehet mondani, hogy DELPHI esetén semmi sem szól amellett, hogy jobban távolodna el az igazi optimumtól, mint egy másik olyan módszer, mely szintén nem veszi figyelembe az igazi célfüggvényt.

Az igazi célfüggvény valamilyen szintű figyelembevétele az első fázisban azt a célt szolgálja, hogy az első fizibilis pont lehetőleg minél jobb legyen az optimalitás szempontjából. Ez a figyelembevétel a belépő változó meghatározásánál történik meg. Szokásos módja a következő.

Legyen z az igazi célfüggvény változó, amely a (2.1) felírásban valamelyik y_i logikai változónak felel meg és amelyet maximalizálni kell, w pedig az infizibilitások (5.2)-ben definiált mértéke. Képezzük ezekkel az

$$s = w + \lambda z \tag{5.46}$$

összetett alakot valamilyen $\lambda > 0$-val és maximalizáljuk s-et. Ez azt jelenti, hogy s szerinti javító vektorokat keresünk. Ha ilyen oszlopot nem találunk, akkor a megoldás s-optimális, amit még további kiértékelés alá kell vetni. Ha egy s-optimális megoldásban $w=0$ teljesül, akkor az eredeti feladat egy optimális megoldásánál

vagyunk (a megoldás z-optimális). Ha $w<0$, akkor a feladat fizibilitását még nem tudjuk eldönteni. Ezért λ-t nullára állítjuk és így gyakorlatilag w szerinti javító vektorokat keresünk, ami egy tiszta első fázist jelent. Ha a feladatnak *nincs fizibilis megoldása,* akkor λ nagyságának előnytelen megválasztása, valamint az a törekvés, hogy, amíg csak lehet, az (5.46)-beli s szerint keresünk javító vektorokat, késleltetheti az infizibilitás felismerését, mert lehetnek olyan oszlopok, melyek w szerint nem javítanak, de z szerint igen és w, λ valamint z relatív nagyságrendje következtében ezek az oszlopok s szerint javítást jelentenek. Ha azonban a feladatnak *van fizibilis megoldása,* akkor a kényszerűségből működtetett tiszta első fázis eljárás során az igazi célfüggvény szerinti előrehaladást részben, vagy teljesen el lehet veszíteni, ami viszont feleslegessé teszi a korábbi iterációk egy részét, ez pedig a hatékonyság csökkenését jelenti. Ezen eljárás esetleges további hátrányainak elemzése helyett pusztán arra a tényre utalunk, hogy az irodalomban rendkívül korlátozott információ áll rendelkezésre ennek a megközelítésnek a sikeréről, noha minden nagyobb LP rendszer fel van szerelve ezzel a lehetőséggel (amivel opcionálisan lehet élni).

A fentiekkel szemben nagyon kedvező számítástechnikai tapasztalatokat sikerült szerezni egy új adaptív módon működő összetett célfüggvényű első fázis eljárással (ADA*ptive* COM*posite* P*rocedure*), mely néhány vonásában hasonlít a most leírthoz, de attól 2 lényeges pontban eltér:

a) egy oszlopot akkor tekintünk javító vektornak, ha w szerint javít és az ilyenek közül választjuk ki azokat, amelyek s szerint is javítanak (ha ilyenek vannak),
b) a λ súlyfaktort dinamikusan változtatjuk az aktuális helyzetnek megfelelően.

Legyen z_j a j-edik oszlop z szerinti árnyékára, és d_j a w szerinti árnyékár, amint azt (5.7)-ben definiáltuk.

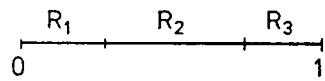
ADACOMP-ot többszörös kiválasztási (MPRICE) környezetben írjuk le. Jelölje N_s az egy major iteráció során kiválasztható javító vektor jelöltek maximális számát. Az iterációk kezdete előtt λ-nak valamilyen $\lambda>0$ kezdeti értéket adunk. Ha egy fő iteráció során több, mint N_s javító vektor van w szerint, akkor egy másodlagos kiválasztási szabály lép életbe. Ez előnyben részesíti azon w szerinti javító vektorokat, melyekre $d_j+\lambda z_j$ negatívabb. Az ezek után következő szuboptimalizálást is csak legfeljebb addig folytatjuk, amíg a minor iterációk során vannak w szerinti javító vektorok.

Egy major iteráció során két számlálót használunk:

— L_1 számlálja a negatív d_j-ket,
— L_2 számlálja azokat a negatív d_j-ket, melyekre még a $d_j+\lambda z_j<0$ is teljesül.

Ha $L_1=0$, akkor a feladat infizibilis, teljesen függetlenül λ aktuális értékétől, vagy találtunk egy fizibilis megoldást.

Ha $L_1\neq 0$, akkor a következő major iterációra λ értékét a következő módon definiáljuk. Legyen $\varrho=L_2/L_1$. Nyilvánvaló teljesül ϱ-ra, hogy $0\leqq\varrho\leqq 1$. A [0, 1] intervallumot három részre osztjuk fel az alábbiak szerint



13. ábra

úgy, hogy $[0, 1]=R_1 \cup R_2 \cup R_3$ és az R_i-k diszjunktak. λ új értékét, melyet $\bar{\lambda}$-val jelölünk, az alábbiak szerint definiáljuk:

$$\begin{aligned} &\text{Ha} \quad \varrho \in R_1, \quad \text{akkor} \quad \bar{\lambda} = g_1(\lambda), \\ &\text{ha} \quad \varrho \in R_2, \quad \text{akkor} \quad \bar{\lambda} = g_2(\lambda) \equiv \lambda, \\ &\text{ha} \quad \varrho \in R_3, \quad \text{akkor} \quad \bar{\lambda} = g_3(\lambda). \end{aligned} \tag{5.47}$$

Itt $g_1(\lambda)$ és $g_3(\lambda)$ olyan függvények, melyekre $0 \leqq \bar{\lambda} = g_1(\lambda) < \lambda$ és $0 \leqq \lambda < \bar{\lambda} = g_3(\lambda)$ teljesül.

(5.47)-et úgy értelmezhetjük, hogy ha a w szerint javító vektorok nagy része s szerint is javít ($\varrho \in R_3$), akkor — ebben a kedvező helyzetben az igazi célfüggvény irányába tett elmozdulást nagyobb súllyal is figyelembe lehet venni, ami λ növelésével érhető el. Az ellenkező esetben, amikor a w szerint javító vektorok nagy része z szerint nagyon ront, az optimalitás kevésbé lesz érdekes a fizibilitással szemben, amit λ csökkentése fejez ki. (5.47) második sora azt jelenti, hogy s két komponense — egy általunk definiált — egyensúlyban van és így nincs szükség λ értékének megváltoztatására.

Könnyen látható, hogy (5.47)-tel nagyfokú flexibilitást hoztunk létre az algoritmus számára. A paraméterek (az R_1, R_2 és R_3 halmazok, λ kezdeti értéke, valamint a g_1 és g_3 függvények) aktuális megválasztása lehetőséget ad arra, hogy a megoldó algoritmus hozzá tudjon hangolódni a konkrétan megoldandó feladathoz. Bármilyen is ezen paraméterek megválasztása, ADACOMP feltétlenül helyes eredményt szolgáltat a következő értelemben:

5.6. TÉTEL. ADACOMP a szabad paraméterei tetszőleges megválasztása esetén is véges számú lépésben vagy talál egy lehetséges megoldást, vagy kimutatja, hogy a feladatnak nincs lehetséges megoldása, amennyiben a degeneráció esete matematikailag korrekt módon van kezelve.

Bizonyítás: A degeneráció matematikailag korrekt kezelése az ismert módszerek valamelyikének (pl. lexikografikus eljárás) esetleges szükség szerinti alkalmazását jelenti.

Ha egy major iteráció során $L_1=0$ adódik, akkor a feladatnak nincs lehetséges megoldása, mert első fázisú major iterációra csak akkor kerül sor, ha az aktuális megoldás még infizibilis és miután $L_1=0$ azt jelenti, hogy nincs w szerint javító vektor, ezért a feladat infizibilis.

Ha $L_1 \neq 0$ minden major iteráció során, akkor nem fordulhat elő bázisismétlődés, mert a minor iterációk is csak a w szerint is javító vektorokon történnek és a degeneráció — a feltevés szerint — helyes kezelésben részesül. Ilyen körülmények közt viszont ADACOMP véges számú lépésben egy lehetséges megoldás megtalálásával tud csak befejeződni.

Meg kell jegyezni, hogy ADACOMP alkalmazása bizonyos többlet számítási munkával jár. Az egyébként is számított d_j értékek mellett szükség lehet a z_j árnyékárakra is. Ez utóbbit az

$$z_j = \pi^T \mathbf{a}_j \tag{5.48}$$

skalár szorzással kaphatjuk meg, ahol π a valódi célfüggvény szimplex szorzója.

π-t a következő módon lehet meghatározni:

$$\pi^T = \mathbf{e}_v^T \mathbf{B}^{-1}. \tag{5.49}$$

Itt $\mathbf{e}_v$-vel jelöltük a célfüggvényhez tartozó m-dimenziós egységvektort. Ez egy BTRAN művelet végrehajtását jelenti, amit párhuzamosan lehet elvégezni a többi, BTRAN-nak alávetett vektorral. Az (5.48) skalár szorzást azonban csak azokra az oszlopokra kell elvégezni, melyeknél szükséges a $d_j+\lambda z_j$ szerinti kiértékelés.

Az ADACOMP eljárással kapcsolatos tapasztalatok szerzése céljából a LIPROS programcsomag DELPHI-vel felszerelt változatába építettem be ADACOMP-ot. A sok feladatot magábafoglaló összehasonlító futások is elsősorban a LIPROS+DELPHI és a LIPROS+DELPHI+ADACOMP között bonyolódtak le. Egy feladat esetében sikerült különböző gépeken levő LP programcsomagokkal összehasonlítani az itt bemutatott eljárást.

A próbafutások a LIPROS rendszernek az R—22 számítógépre adaptált változatával történtek multiprogramozási környezetben, a gépnek különböző terhelési feltételei közt. Így a futások CPU idői nem fejezik ki teljesen a tényleges számítási munkát (lásd 3.1. pont), de bizonyos tendenciákat azért mutatnak.

A futások során ADACOMP szabad paraméterei az alábbi értékeket kapták:

— λ induló értéke: $\lambda=0{,}5$,
— $R_1=[0, 1/3)$; $R_2=[1/3, 2/3)$; $R_3=[2/3, 1]$,
— $g_1(\lambda)=\lambda/2$; $g_3(\lambda)=2\lambda$.

A következő táblázaton 3, többé-kevésbé tipikusnak talált feladat 3 különböző N_s érték melletti futási statisztikáját mutatjuk be. A feladatok mérete $m\times n$ formában van megadva. IT jelöli a megoldás eléréséhez szükséges össz iteráció számot, míg IT(PH-I) az iterációk számát az első fázisban. CPU-val a processzor időt jelöltük. D=DELPHI; D+A=DELPHI+ADACOMP.

Az egyik legfontosabb tapasztalat az, hogy ADACOMP használata esetén mindig lényegesen kevesebb lépés maradt a második fázisra, vagyis az ADACOMP-pal kapott első fizibilis megoldás — az elvárásokkal összhangban — határozottan közelebb adódik az optimumhoz, mint az ADACOMP nélküli (D) esetekben. Ezzel egyidőben az összes iterációk száma is lényegesen, a bizonytalanabbul meghatározott CPU idő pedig hol nagyon, hol kevésbé csökkent, és csak egy esetben — ott is a mérési hiba határain belül — növekedett egy kicsit a D esethez képest. Nagyobb feladatokkal végzett teszt futások szintén alátámasztják ezt a megfigyelést és gyakran még az itt észleltnél is nagyobb megtakarítást mutattak az iterációk számában. Miután ilyen feladatokat csak alkalomszerűen oldottunk meg (nagy feladatokkal való kísérletezés túl sok gépidőt fogyaszt, így túl drága), ezért az előzőekhez hasonló szisztematikus tapasztalatok nem állnak rendelkezésre. Az ADACOMP működésére vonatkozó néhány részletes statisztikai adat a Függelék II. pontjában található.

Még nehezebb a helyzet, ha különböző gépeken működő LP rendszerekkel akarunk összehasonlító futásokat csinálni. Itt a különböző rendszerek használatának az ismeretétől kezdve az adatok hibátlan átvitelén keresztül a futások azonos — tehát összehasonlítható — körülményeinek a megteremtéséig számos nehézség merül fel. Más intézetekben dolgozó néhány kolléga szíves segítségével azonban sikerült a 62×70-es feladatot néhány különböző nagy programcsomaggal, azonos feltételek között, N_s=4-gyel megoldani. Miután a számítógépek különböző típu-

	$N_s = 4$		$N_s = 6$		$N_s = 8$	
	D	D+A	D	D+A	D	D+A
1. Feladat 41×60						
IT	76	64	79	54	66	37
IT (PH – I)	49	57	36	47	42	34
CPU	3'28,9"	3'03,8"	3'33,1"	2'16,3"	2'37,7"	1'37,5"
2. Feladat 62×70						
IT	55	30	45	31	35	31
IT (PH – 1)	34	29	37	29	31	29
CPU	1'33,8"	1'07,0"	1'09,6"	1'08,3"	0'59,8"	1'04,3"
3. Feladat 100×130						
IT	136	83	145	84	125	88
IT (PH – 1)	74	75	75	62	76	76
CPU	7'29,0"	3'36,9"	6'55,5"	4'12,1"	5'46,8"	4'15,9"

súak voltak, ezért az alábbi táblázatban csak a teljes megoldáshoz szükséges iterációk számát tüntetjük fel:

LPS (IBM/DOS)	81
MPS (IBM/OS)	56
LIPROS+DELPHI	55
LP—400 (ICL)	42
LIPROS+DELPHI+ADACOMP	30.

Természetesen helytelen volna levonni azt a következtetést, hogy a LIPROS+D+A ennyivel jobb lenne a többi programcsomagnál. Azt azonban látni lehet, hogy egy teljesen találomra kiválasztott feladat esetén igen nagy javulás adódott, és talán ez is bátorítást adhat a DELPHI+ADACOMP részleteinek finomabb kidolgozására és egy komolyabb teljesítményvizsgálat elvégzésére.

5.3. *CRASH*

A CRASH [27] eljárás régóta ismeretes és használatos induló bázis keresési technika a lineáris programozás gyakorlatában. Lényege az, hogy a tisztán logikai vektorokból álló bázis helyett egy trianguláris bázist állít elő, melyben lehetőleg minél több strukturális oszlop szerepel. A triangularitás kulcsponti szerepe a 2.3.1. lemma alapján válik érthetővé. Az ilyen bázis ugyanis a bázisba belépő vektorok transzfor-

mációja nélkül keletkezik, így sűrűsége minimális, pontossága ugyanakkor a lehető legnagyobb. Ezek eredményeképpen a CRASH-t követő szimplex iterációs lépések gyorsak és pontosak.

A CRASH eljárásoknak igen sok változata ismeretes, mégis nagyjából 2 fő csoportba sorolhatók:

a) törekvés a minél nagyobb trianguláris bázis megtalálására a strukturális változók körében,
b) a trianguláris bázis keresése közben a fizibilitás és/vagy optimalitás szempontjának figyelembevétele.

Jelen pontban ezt a listát egy új szemponttal bővítjük:

c) triangularitás mellett törekvés arra, hogy a bázis degeneráltsági foka csökkenjen.

Ezután a *b)* alatti szempontokat is újraértékeljük DELPHI, illetve DELPHI-A ismeretében.

A gyakorlatban igen sűrűn fordulnak elő olyan feladatok, melyek jobboldala sok 0 elemet tartalmaz, így a tisztán logikai vektorokból álló bázis meglehetősen degenerált. Az ilyen báziról való előrelépés nehézségei köztudottak.

A CRASH eljárás a tisztán logikai vektorból álló bázisból indul ki és ennek oszlopait cseréli ki olyan strukturális oszlopokra, melyek trianguláris bázist alkotnak (ha ilyet egyáltalán lehet csinálni). A logikai bázis esetén

$$\boldsymbol{\alpha}_j = \mathbf{B}^{-1}\mathbf{a}_j = \mathbf{I}\mathbf{a}_j = \mathbf{a}_j. \tag{5.50}$$

A degeneráció mértékének a csökkenése akkor következhet be, amikor olyan báziscserét hajtunk végre, melynél a belépő változó nullától (ill. felsőkorláttól) különböző értéket vesz fel. Ehhez az szükséges, hogy a megfelelő pivot sorra $\beta_i \neq 0$, és a belépő vektor megfelelő helyén $\alpha_{ij} \neq 0$ teljesüljön. Ilyen oszlopok keresésénél a kritikus pont az α_{ij} értékek ismerete (vö. 5.6. és 5.7. lemma). A CRASH technika alkalmazása esetén azonban a keresésből ki nem zárt oszlopok olyanok, hogy 0 elemet tartalmaznak az összes eddigi pivot sorban, így a 2.3.1. lemma értelmében egyetlen eddigi elemi transzformációs mátrixszal sem kell azokat transzformálni, így a keletkező nem triviális bázissal is igaz, hogy ezekre az oszlopokra

$$\boldsymbol{\alpha}_j = \mathbf{B}^{-1}\mathbf{a}_j = \mathbf{a}_j,$$

tehát az $\alpha_{ij} = a_{ij}$ értékek ezekben az oszlopokban ismertek.

Ezek előrebocsátásával az alábbi degeneráció ellenes CRASH eljárás adódik.

Legyen K_i az i-edik bázisváltozó eredeti indexe (2.5)-ben, továbbá legyen

$$H_2 = \{i\colon \beta_i \neq 0 \wedge K_i \leqq m\} \tag{5.51}$$

β_i kezdetben $\bar{b}_i$-vel egyenlő (vö. 2.6b képlet). $K_i \leqq m$ jelenti, hogy még nem történt pivotálás az i-edik soron.

1. Végignézni a mátrix még szabad oszlopait (kezdetben ez az összes oszlop) és számolni rájuk a

(5.52) $$c_j = \sum_{\substack{i \in H_2 \\ a_{ij} \neq 0}} 1 \quad \text{(illetve } c_j = 0, \text{ ha a szummációs halmaz üres)}$$

értéket.
Ha már nincs szabad oszlop, vagy $\max c_j = 0$, akkor az eljárás véget ér. Ellenkező esetben kiválasztjuk azt az oszlopot, melyre c_j maximális. Ezzel igyekszünk olyan oszlopot találni, melynek minél több potenciális nemdegenerált pivot sora van, ami által lehetőség nyílik további szempontok érvényesítésére. Ha a $\max c_j$ több oszlopra is felvétetik, akkor célszerű azt az oszlopot kiválasztani, melyben több nem-nulla elem van, mert egy ilyen oszlop bevonása jobban megváltoztatja a jobboldalt.

2. A kiválasztott oszlopra definiáljuk a H_3 indexhalmazt:

$$H_3 = \{i: i \in H_2 \wedge a_{ij} \neq 0\}$$

és képezzük a

$$T_i = \beta_i / a_{ij}, \quad i \in H_3$$

hányadosokat. A T_i-k a belépő változó különböző nagyságát határozzák meg abban az esetben, ha a megfelelő kilépő változó 0 szinten hagyja el a bázist. (Felső korlátos logikai változók esetén természetesen definiálni lehetne a DELPHI-nél szereplő T_{u_i} típusú hányadosokat is.) Az $i \in H_3$ sorok mind olyanok, amelyekhez nem-degenerált báziscsere tartozik. Ezek közül további szempontok alapján lehet a pivot sort kiválasztani mint például:
— törekvés a fizibilitás javítására (a H_3-beli pontokra korlátozott DELPHI-A értelemszerű alkalmazásával),
— minimális sorszámláló, hogy minél kevesebb további oszlopot zárjunk ki a későbbi jelöltek köréből.

3. Az újonnan keletkezett elemi transzformációs mátrixszal transzformálni a jobboldalt, újra definiálni a H_2 halmazt (ami a transzformáció során megváltozhat, az eljárás célkitűzésével összhangban), elvégezni a hagyományos CRASH technika adminisztrációs lépéseit és visszatérni az 1. lépésre.

Ha ez a 3 lépéses ciklus elakad, meg lehet kísérelni, hogy olyan 0 jobboldalú soron pivotoljunk, melyhez 0 típusú logikai változó tartozik („egyenlőség" típusú feltétel). Egy ilyen pivot lépés során ugyanis egy 0 típusú változó lép ki a bázisból, ami azért előnyös, mert egy ilyen változó a későbbi iterációk során könnyen akadályt jelentene, ugyanis bármerre elmozdulva infizibilissé válna, és ha mellette nagy $|\alpha_i|$ áll, akkor ez hamar okozná a megfelelő $w(t)$ maximumának elérését DELPHI használata esetén; ha viszont erre a helyre a bázisba egy nem 0 típusú változó kerül nulla szinten, akkor a későbbiekben ennek legalább az egyik irányú elmozdulása fizibilis, így jó előjelű α_i esetén nem vesz részt a megfelelő $w(t)$ függvényben, mert nem definiál töréspontot (lásd DELPHI részletei), ezért nem is idézi elő $w(t)$ „korai" maximumát.

Természetesen azt is meg lehet tenni, hogy eleve megengedünk ilyen típusú lépéseket. Ez esetben is alkalmazhatjuk az előbbi 3 lépéses eljárást olyan módosítással, hogy az ott szereplő H_2 halmaz helyett a

$$H_2' = \{i\colon (i \in (I_0 \cap F) \vee \beta_i \neq 0) \wedge K_i \leqq m\}$$

halmazzal dolgozunk és ekkor nem lesz igaz, hogy minden $i \in H_3$ sorhoz nemdegenerált báziscsere tartozik, de azokat ilyenkor is preferálni lehet, amíg egyáltalán léteznek.

Amennyiben nem ezt a degeneráció elleni CRASH technikát alkalmazzuk, hanem egyszerűen csak egy *b)* típusú eljárást akarunk használni, akkor is nagyon előnyösen lehet használni DELPHI-t illetve DELPHI-A-t a szabad pivot sorok közötti választásra.

IRODALOM

[1] BARTELS, R. H. and GOLUB, G. H., "The simplex method of linear programming using LU decomposition", *Comm. A.C.M.* **12** (1969) 266—268.

[2] BEALE, E. M. L., "The current algorithmic scope of mathematical programming systems", *Mathematical Programming Study* **4** (1975).

[3] BENICHOU, M. és társai, "The efficient solution of large-scale linear programming problems", *Math. Progr.* **13** (1977).

[4] CROWDER, H. P. and HATTINGH, J. M., "Partially normalized pivot selection in linear programming", *Math. Progr. Study* **4** (1975).

[5] DANTZIG, G. B., "Maximization of a linear function of variables subject to linear inequalities", *T. C. Koopmans* (ed.)*: Activity analysis and production allocation* (Wiley, New York, 1951).

[6] DOBOSY, A., HEPPES, A. és MAROS, I., „Nagyméretű LP feladatok megoldása ICL 1900-as gépeken", *Szervezés és Vezetés* **11** (1974).

[7] GRAVES, R. L. and WOLFE, P. (eds.), *Recent Advances in Mathematical Programming* (McGraw-Hill, 1963).

[8] GREENBERG, H. J., "A tutorial on matricial packing", Summer school on design and implementation of optimization software, Urbino, 1977.

[9] HADLEY, G., *Linear Programming* (Addison-Wesley, 1962).

[10] HARRIS, P. M. J., "Pivot selection methods of the devex LP code", *Math. Progr.* **5** (1973).

[11] HELLERMAN, E. and RARICK, D., "Reinversion with the preassigned pivot procedure", *Math. Progr.* **1** (1971) 195—216.

[12] HELLERMAN, E. and RARICK, D., "The partitioned preassigned pivot procedure (P^4)", *Sparse Matricies and Their Applications* eds. *D. J. Rose & R. A. Willoughby*, Plenum Press, 1972.

[13] HEPPES, A. és MAROS, I., „Program nagyméretű lineáris programozási feladatok megoldására", *SZÁMOLÓGÉP* **3** (1971).

[14] JACKSON, R. and HOFFMAN, K., "The testing and evaluation of mathematical programming software", EURO IV, Cambridge, 1980.

[15] KALAN, J. E., "Aspects of large-scale in-core linear programming", Proceedings of the 1971 annual conference of the ACM (Chicago, 1971).

[16] KÉRI, G., „Lineáris programozási technikák: 1. A lineáris programozási feladatok kitűzési módjai", Working paper, MTA SZTAKI, 1979.

[17] MAROS, I., „Nagyméretű LP feladatok megoldására szolgáló FUT rendszer II", NIM IGÜSZI 1221.01/1971.

[18] MAROS, I., „Hibaeljárások lineáris programozási algoritmusokban", *Információ-Elektronika* **2** (1974).

[19] MAROS, I., „Adaptív elemek a lineáris programozásban", *Alk. Mat. Lapok* **2** (1976).

[20] MAROS, I., MÓCSI, Z.-NÉ, NAGY, P.-NÉ és SZILÁGYI, P., „Az LP2 lineáris programozási programcsomag tesztelése", SZÁMKI 1977/78.

[21] MAROS, I. and MÓCSI, J., "Investigation of the numerical behavior of a dual type linear programming algorithm", SZÁMKI TANULMÁNYOK 1978/2.

[22] MAROS, I. and MÓCSI, J., "Experiences with the dual type GUB algorithm of Grigoriadis", *Prékopa, A.* (ed.) *Survey of Mathematical Programming*, I, North-Holland, Akadémiai Kiadó, 1979.

[23] MAROS, I., „Dinamikus normálás a LIPROS lineáris programozási programcsomagban", *Információ-Elektronika* **5** (1979).
[24] MAROS, I., „Lineáris programozási programcsomag R10/12 gépekre", *Struktúra* **9** (1979).
[25] MAROS, I., „A bázisból kilépő vektor meghatározásának egy módja a szimplex módszer első fázisában", *Alk. Mat. Lapok* **6** (1980).
[26] MAROS, I., "A non-standard Phase-I. method", SZÁMKI TANULMÁNYOK 1980/7.
[27] ORCHARD-HAYS, W., *Advanced Linear Programming Computing Techniques* (McGraw-Hill, 1968).
[28] PRÉKOPA, A., *Lineáris programozás*, I (Bolyai János Matematikai Társulat kiadványa, 1968).
[29] RALSTON, A., *Bevezetés a numerikus analízisbe* (Műszaki Kiadó, 1969).
[30] IBM 4341 Processor Functional Characteristics.
[31] IBM Mathematical Programming System Extended 370 (MPSX/370) Program Reference Manual.

(Beérkezett: 1981. április 27.)

MAROS ISTVÁN
SZÁMÍTÓGÉPALKALMAZÁSI KUTATÓINTÉZET
1536 BUDAPEST, PF. 227

ADAPTIVITY IN LINEAR PROGRAMMING II.

I. MAROS

The capabilities (reliability, efficiency, etc.) of the present day LP packages could well be enhanced by a supervising algorithm which could be in charge of properly setting the parameters of the package and thereby controlling the progress of optimization. The paper discusses the details of the demand for such an algorithm, and outlines the role that can be played by adaptivity in fulfilling it. The second part of the paper presents some new adaptive algorithmic techniques to help to overcome the difficulties caused by infeasibility and degeneracy. In the case of two new and implemented procedures (DELPHI and ADACOMP) report on quite favourable computational experiences is also given.

FÜGGELÉK

I, *DELPHI működésére vonatkozó néhány statisztikai adat*

A DELPHI eljárás működését jól jellemzi az, hogy összesen hány töréspontot definiál (Q) és ezek közül hányat használ fel $w(t)$ maximalizáláshoz (q). Az alábbi statisztikában DELPHI lokális viselkedését mutatjuk be különböző bázisokon. A bázisok két feladatból valók. Tekintettel arra, hogy a hagyományos pivotválasztási módszernek a $q=1$ eset felel meg, az alábbi statisztikából azt is látni lehet, hogy az egyes iterációs lépéseknél hogyan viszonyul egymáshoz a hagyományos módszer és a DELPHI eljárás.

Először az adatokat táblázatos formában mutatjuk be, majd az infizibilitások számának alakulását mindkét feladat esetén külön grafikonon is ábrázoljuk. A vízszintes tengelyen a bázisok sorszáma (IB), a függőleges tengelyen az infizibilitások száma (INF) van felmérve.

A) FELADAT

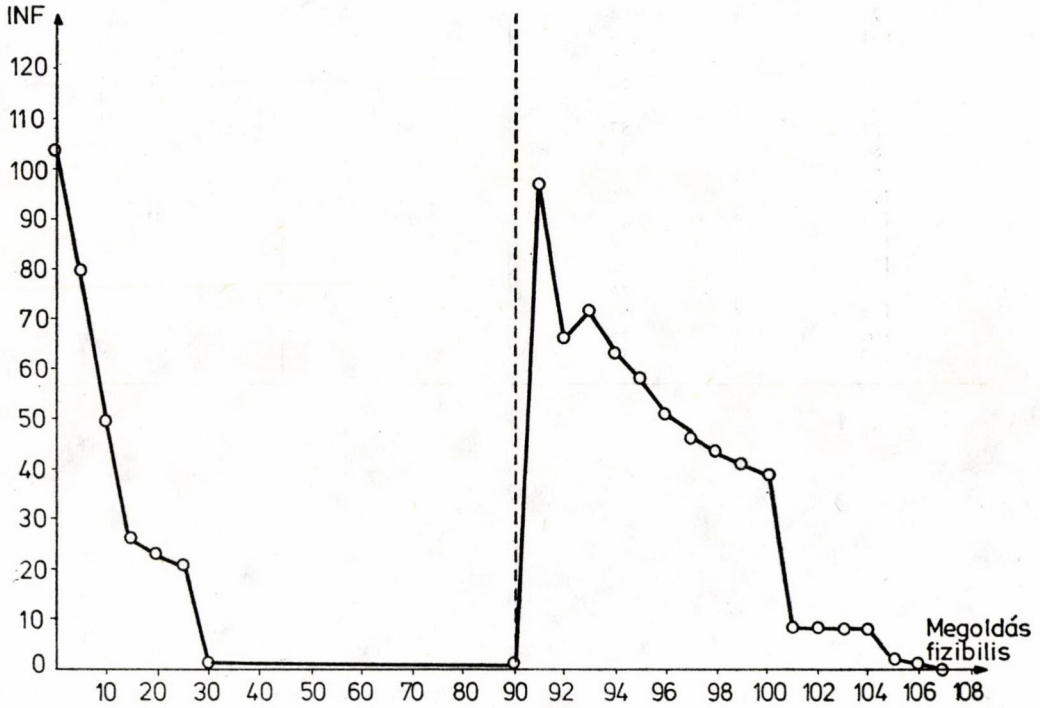
Méret: 237×184 (azonos az 5.1.2. pont táblázatának 5. feladatával)

Bázis száma (IB)	Q	q	Infizibilitások száma	Megjegyzés
1	44	16	104	
2	128	41	101	
3	9	9	95	
4	78	4	88	
5	125	8	80	
6	87	5	76	
7	86	4	78	
8	198	34	63	
9	93	37	50	
10	1	1	49	
20	132	5	23	
30	93	54	1	
40	100	49	1	
50	84	25	1	
60	110	2	1	degenerált
70	87	27	1	lépések
80	95	5	1	
90	71	41	1	
91	102	98	97	!
92	162	51	66	!
93	95	37	72	!
100	2	2	39	
103	16	1	8	
105	9	6	2	
106	1	1	1	
107	2	1	0	

A) FELADAT (erősen degenerált feladat):

méret: 237×184;
felsőkorlátos változók száma: 55;
vektorok száma a szubopt.-ban, $N_s=4$;
iterációk száma az első fázisban: 107.

(A 90. bázisnál léptékváltás van az IB tengelyen, hogy pontosabban lehessen ábrázolni az infizibilitások számának változását.)



14. ábra

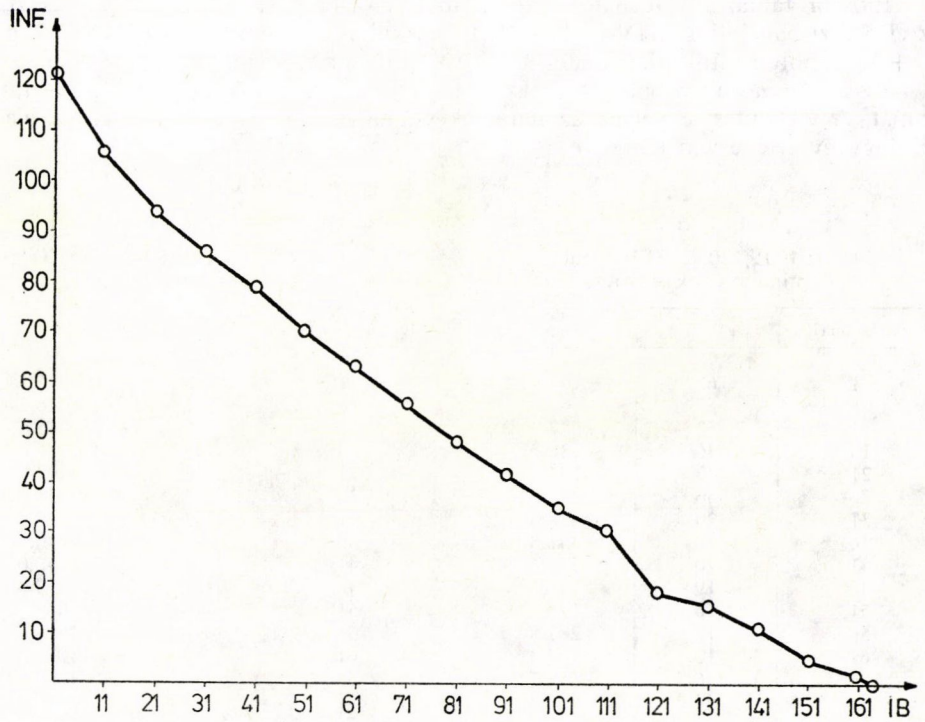
B) FELADAT

Méret: 150×200, sűrűség: 3,7%, 100 egyenlőség típusú feltétel, nincs egyedi felsőkorlátos változó

Bázis száma (IB)	Q	q	Infizibilitások száma
1	4	4	121
11	6	3	106
21	4	2	94
31	8	4	86
41	8	3	79
51	10	5	70
61	59	3	63
71	57	1	56
81	57	3	48
91	9	3	42
101	76	2	35
111	80	10	31
121	74	1	19
131	76	2	17
141	59	6	12
151	60	2	5
161	58	2	2
163	51	1	0

B) FELADAT:

méret: 150×200;
felsőkorlátos változók száma: 0;
egyenlőség feltételek száma: 100;
sűrűség: 3,7%;
vektorok száma a szubopt.-ban, $N_s = 8$;
iterációk száma az első fázisban: 163.



15. ábra

II. *ADACOMP működésére vonatkozó néhány összehasonlító statisztikai adat*

Igen tanulságos annak a vizsgálata, hogy hogyan alakul az igazi célfüggvény értéke az első fázisbeli iterációk során. Ennek érzékeltetésére két összehasonlító futás részleteit mutatjuk be, melyeket két különböző feladat megoldása során szereztünk.

Először táblázatos formában adjuk meg a célfüggvény értékének alakulását az első fázisban külön a DELPHI-vel (D) és külön a DELPHI+ADACOMP-pal (D+A) kapott adatok alapján, ezután egyesített ábrán szemléltetjük az adatokat.

Az ábrán vastag vonal jelöli a D futás, míg vékony vonal a D+A futás grafikonját. A vízszintes tengelyen az iterációk száma, a függőleges tengelyen az igazi célfüggvény értéke van felmérve.

A) FELADAT

Méret: 100×130, DELPHI futás
optimum érték: 1141,1

Iterációszám (IB)	INF	Célfüggvény
1	65	104,0
6	57	−65,5
11	52	−637,9
16	46	−636,7
21	40	−184,1
26	36	−277,0
31	29	−338,3
36	26	−441,8
41	20	−516,6
46	19	−733,3
51	16	−176,2
56	12	−287,8
61	10	−127,3
66	6	−543,1
71	4	−560,4
76	0	−581,7

A második fázisban még 49 iterációra voltszükség egy optimális megoldás eléréséig.

A) FELADAT

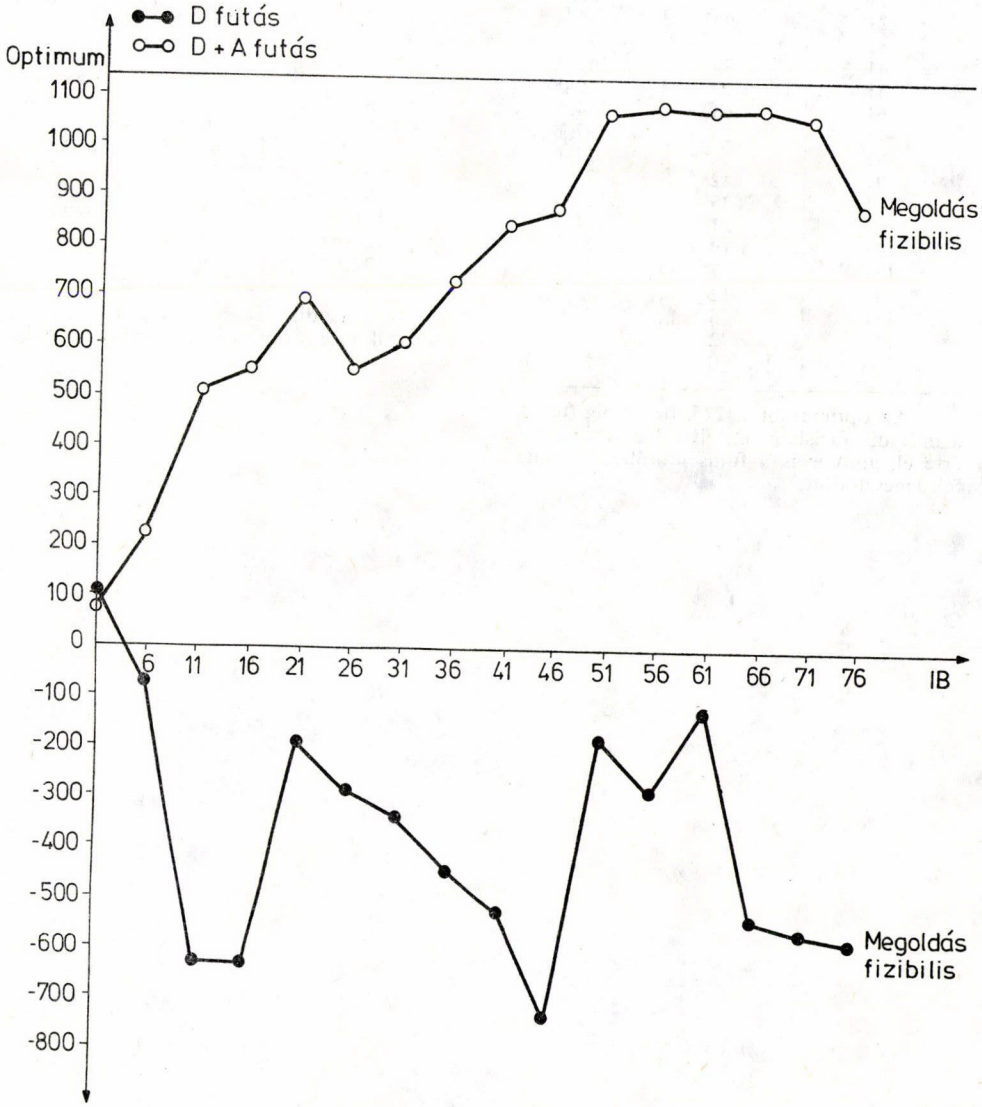
Méret: 100×130, DELPHI+ADACOMP
futás optimum értéke: 1141,1

Iterációszám (IB)	INF	Célfüggvény
1	68	75,1
6	61	226,2
11	55	515,6
16	49	551,6
21	43	697,8
26	37	553,1
31	35	617,9
36	29	730,7
41	24	848,0
46	20	879,0
51	14	1069,8
56	8	1086,7
61	3	1075,1
66	1	1077,6
71	1	1009,0
76	0	871,7

A második fázisban még 12 iterációra volt szükség egy optimális megoldás eléréséig.

A) FELADAT:

méret: 100×130;
egyenlőség feltételek száma: 50, $N_s = 8$;
iterációk száma az első fázisban: D: 76
D+A: 76;
iterációk száma a második fázisban: D: 49
D+A: 12.



16. ábra

B) FELADAT

Méret: 150×200, DELPHI futás
optimum érték: 819,0

Iterációszám	INF	Célfüggvény
1	121	24,9
11	106	66,9
21	94	56,7
31	86	75,8
41	79	161,9
51	70	249,0
61	63	284,5
71	56	207,3
81	48	137,4
91	42	72,5
101	35	−114,8
111	31	−1,4
121	19	24,8
131	17	43,9
141	12	78,5
151	5	−43,6
161	2	10,4
163	0	19,1

Az optimumot a 225. iterációig (tehát a második fázisban 62-t iterálva) még nem érte el, amikor is a futás időtúllépés miatt felfüggesztődött.

B) FELADAT

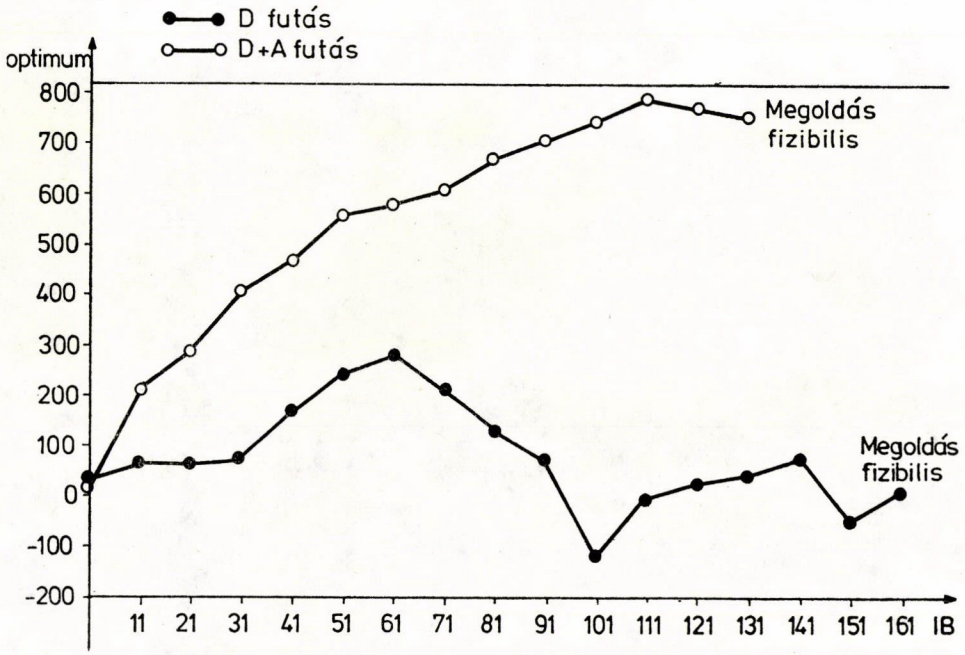
Méret 150×200, DELPHI+ADACOMP
futás optimum érték: 819,0

Iterációszám	INF	Célfüggvény
1	120	18,6
11	109	213,9
21	100	292,9
31	86	405,5
41	78	472,6
51	71	553,6
61	61	581,3
71	50	617,6
81	44	672,2
91	36	717,1
101	25	755,3
111	21	797,4
121	10	781,1
131	0	769,2

A második fázisban még 4 iterációra volt szükség egy optimális megoldás eléréséig.

B) FELADAT:

méret: 150×200;
egyenlőség feltételek száma: 100;
$N_s = 4$;
iterációk száma az első fázisban: D: 163
D+A: 131;
iterációk száma a második fázisban: D: (62 után még nincs megoldás)
D+A: 4.



17. ábra

Alkalmazott Matematikai Lapok 7 (1981) 73—82

DÖNTÉSEKTŐL FÜGGŐ ELLÁTÁSI FELADATOK MEGOLDÁSA BENDERS DEKOMPOZÍCIÓVAL

HOFFER JÁNOS

Budapest

A dolgozat a döntésektől függő ellátási feladatok *Benders dekompozícióval* történő megoldásakor fellépő lineáris programozási feladatokkal, illetve azok megengedett tartományának extremális pontjainak és irányainak előállításával foglalkozik. A felvetett kérdések megválaszolására egyszerű eljárást adunk az úgynevezett „mohó" algoritmus és a kiegészítő eltérések tételének felhasználásával.

1. A feladat kitűzése és a Benders dekompozíció alkalmazása

A dolgozatban ellátási feladatnak nevezzük azoknak a — köznapi életben igen gyakran felmerülő — gyakorlati feladatoknak matematikai megfogalmazását, amelyek egy termék előírt mennyiségének több munkahelyen történő lehető legolcsóbb előállítását célozzák. Amennyiben az egyes munkahelyek gyártóképessége és az előállítandó összmennyiség döntésektől függ, döntésektől függő ellátási feladatokról beszélünk. Formulákkal megfogalmazva a következő típusú lineáris, vegyes, egészértékű programozási feladatokkal foglalkozunk:

$$
\begin{aligned}
& x_1 + \quad \mathbf{f}_1'\mathbf{y} && \leqq b_1, \\
& x_2 + \quad \mathbf{f}_2'\mathbf{y} && \leqq b_2, \\
& \qquad \vdots \\
& x_n + \quad \mathbf{f}_n'\mathbf{y} && \leqq b_n, \\
& x_1 + \ldots + x_n + \mathbf{f}_{n+1}'\mathbf{y} = b_{n+1}, \\
& x_1 \geqq 0, \ldots, x_n \geqq 0, \ \mathbf{y} \in \{0, 1\}^m, \\
& \min (\mathbf{c}'\mathbf{x} + \mathbf{d}'\mathbf{y}).
\end{aligned} \tag{1.1}
$$

(1.1)-ben x_i jelöli az i-edik munkahelyen előállítandó termékmennyiséget, y_j pedig a j-edik döntés eredményét.

Ezzel a feladattípussal találkozott a szerző a magyar villamosenergiarendszer napi optimális menetrendjének meghatározásakor [2]. Naponta mintegy 50—100 döntésektől függő ellátási feladatot kell megoldani a menetrend előállításához (ezek mérete: $18 \leqq n \leqq 35$, $m \leqq 21$).

Ha az (1.1) feladatot *Benders dekompozícióval* oldjuk meg (részletesebb leírását lásd [1], [3], [4]), iteratív eljárást kell végrehajtanunk. Ennek egy iterációjában egy egészértékű és egy lineáris programozási feladatot kell megoldani. A dolgozat további részeiben ez utóbbit részletekbe menően tárgyaljuk.

Az említett lineáris programozási feladat nem más, mint az (1.1) feladat duálisa, az egészértékű változóknak valamely rögzítése esetén. A rögzítés mikéntjét az azonos iterációban megoldott egészértékű feladat optimális megoldásvektora szolgáltatja.

Formulákkal megfogalmazva: ha $\mathbf{y}^i$ jelöli az i-edik iterációban megoldott egészértékű feladat optimális megoldását és $u_1, u_2, \ldots, u_{n+1}$ a duális feladat változóit, akkor a megoldandó lineáris programozási feladat a következő:

$$
\begin{aligned}
&u_1+u_{n+1} \geqq -c_1,\\
&u_2+u_{n+1} \geqq -c_2,\\
&\qquad\vdots\\
&u_n+u_{n+1} \geqq -c_n,\\
&u_1 \geqq 0, \ldots, u_n \geqq 0\\
&\mathbf{d}'\mathbf{y}^i - \min\left((b_1-\mathbf{f}_1'\mathbf{y}^i)u_1+(b_2-\mathbf{f}_2'\mathbf{y}^i)u_2+\ldots+(b_{n+1}-\mathbf{f}_{n+1}'\mathbf{y}^i)u_{n+1}\right).
\end{aligned}
\tag{1.2}
$$

Bevezetve a

$$
b_j-\mathbf{f}_j'\mathbf{y}^i = g_j, \quad j=1, 2, \ldots, n+1 \tag{1.3}
$$

jelölést, az (1.1) feladat a döntés-változók rögzítése esetén ($\mathbf{y}=\mathbf{y}^i$) a következő:

$$
\begin{aligned}
x_1 &\leqq g_1,\\
x_2 &\leqq g_2,\\
&\vdots\\
x_n &\leqq g_n,\\
x_1+x_2+\ldots+x_n &= g_{n+1},\\
x_1 \geqq 0, \ldots, x_n &\geqq 0,\\
\mathbf{d}'\mathbf{y}^i + \min\,&(c_1x_1+\ldots+c_nx_n).
\end{aligned}
\tag{1.4}
$$

Az (1.4) feladat duálisa, vagyis az (1.2) egyszerűbb formában:

$$
\begin{aligned}
&u_1+u_{n+1} \geqq -c_1,\\
&u_2+u_{n+1} \geqq -c_2,\\
&\qquad\vdots\\
&u_n+u_{n+1} \geqq -c_n,\\
&u_1 \geqq 0, \ldots, u_n \geqq 0,\\
&\mathbf{d}'\mathbf{y}^i - \min\,(g_1u_1+g_2u_2+\ldots+g_{n+1}u_{n+1}).
\end{aligned}
\tag{1.5}
$$

Két megjegyzéssel élünk:

1. Az (1.5) feladatban az u_{n+1} változóra *nincs* nemnegativitási feltétel, lévén az (1.1) feladat utolsó feltétele egyenlőséget követel meg.

2. Bár az (1.3)-ban bevezetett g_j mennyiségek függenek az $\mathbf{y}^i$ vektortól, ezt a tényt itt azért nem tüntettük fel, mert egy feladattípussal foglalkozunk. Az i-edik iterációban megoldandó lineáris programozási feladatok pedig éppen (1.5) típusúak, csupán célfüggvényegyütthatóikban különböznek egymástól.

2. A megoldandó lineáris programozási feladat elemzése és megoldása

A *Benders dekompozíció* egyes iterációiban tehát az (1.5) típusú feladat optimális megoldására van szükségünk, az optimum értékén kívül kíváncsiak vagyunk a feladat megengedett tartományának legalább egy optimális csúcsára, vagy nem korlátos esetben egy negatív célfüggvényértékű extremális irányára.

Az (1.5) feladat helyett annak duálisát, az (1.4) feladatot oldjuk meg; lévén az egy nem korlátos, folytonos változójú hátizsák feladat, melyre jól ismert hatékony algoritmus a *„mohó" algoritmus* (egészértékű esetre vonatkozóan lásd: KOVÁCS L. B. [4]). Az így kapott megoldásból a kiegészítő eltérések tételének segítségével előállítjuk az (1.5) feladat extremálisait.

Megjegyezzük, hogy az (1.5) feladatnak mindig van megengedett megoldása, például a következő:

$$u_{n+1} = 0$$

$$u_i = \max(-c_1, -c_2, \ldots, -c_n, 0), \quad i = 1, \ldots, n.$$

Fogjunk hozzá az (1.4) feladat megoldásához. A feltételek jobb oldala szerint több eset lehetséges.

1. eset

Van $g_i < 0$ ($i \in \{1, 2, \ldots, n\}$). Ekkor (1.4)-nek nincs megengedett megoldása. Minthogy (1.5) megengedett tartománya nem üres, a dualitás tétel következtében (lásd PRÉKOPA [5]) az (1.5) feladat célfüggvénye nem korlátos; az i-edik egységvektor: $\mathbf{e}_i (\mathbf{e}_i \in \mathbf{R}^{n+1})$ negatív célfüggvényértékű megengedett iránya, és mint az könnyen belátható extremális iránya is.

2. eset

$g_i \geqq 0$ ($i = 1, 2, \ldots, n$), $g_{n+1} < 0$. Ekkor az 1. esetben leírt gondolatmenettel azt kapjuk, hogy $\mathbf{e}_{n+1}$ negatív célfüggvényértékű megengedett iránya (1.5)-nek, és az előzőtől némiképp eltérő módon, de könnyen belátható, hogy extremális iránya is.

3. eset

$g_i \geqq 0$ ($i = 1, 2, \ldots, n$), $g_{n+1} = 0$. Ekkor az (1.4) egyetlen optimális megoldása: $\mathbf{x} = \mathbf{0}$, az optimum: $\mathbf{d}'\mathbf{y}^i$, és könnyen belátható, hogy a

$$c = \max(-c_1, -c_2, \ldots, -c_n)$$

jelölést bevezetve, az (1.5) feladat megengedett tartományának optimális csúcsa a $c \cdot \mathbf{e}_{n+1}$ vektor.

A további esetekben $g_i \geqq 0$ ($i = 1, 2, \ldots, n$) és $g_{n+1} > 0$. Rendezzük sorba az (1.4) feladat változóit a következő módon. Álljanak elöl azok a változók, amelyek

felső korlátja nulla; vagyis legyen j az az index, amelyre

$$g_1 = g_2 = \ldots = g_j = 0,$$

és

$$g_{j+1} > 0,\ g_{j+2} > 0,\ \ldots,\ g_n > 0.$$

A többi változót pedig oly módon rendezzük át, hogy célfüggvényegyütthatóik növekvő sorrendben álljanak:

$$c_{j+1} \leqq c_{j+2} \leqq \ldots \leqq c_n.$$

A 4. és 5. esetben feltesszük, hogy

$$g_{n+1} \leqq g_1 + g_2 + \ldots + g_n.$$

4. esetként vizsgáljuk azt az esetet, amikor van olyan k index ($k \leqq n$), amelyre

$$g_1 + g_2 + \ldots + g_k = g_{n+1},$$

5. esetként, amikor olyan k index létezik csak, amelyre

$$g_1 + g_2 + \ldots + g_{k-1} < g_{n+1} < g_1 + g_2 + \ldots + g_k.$$

Végül 6. esetként tárgyaljuk a

$$g_1 + g_2 + \ldots + g_n < g_{n+1}$$

esetet.

4. eset

$g_1 = g_2 = \ldots = g_j = 0;\ g_{j+1} > 0, g_{j+2} > 0, \ldots, g_n > 0;\ g_1 + g_2 + \ldots + g_k = g_{n+1}\ (j < k \leqq n).$

Ekkor az (1.4) feladat optimális megoldása, melyet a c_j együtthatók rendezésével a mohó algoritmus szolgáltat, a következő:

$$x_1 = x_2 = \ldots = x_j = 0, \tag{2.1}$$

$$x_{j+1} = g_{j+1},\ x_{j+2} = g_{j+2},\ \ldots,\ x_k = g_k, \tag{2.2}$$

$$x_{k+1} = x_{k+2} = \ldots = x_n = 0. \tag{2.3}$$

A kiegészítő eltérések tétele (PRÉKOPA [5]) alapján az (1.5) minden $u_1, u_2, \ldots, u_{n+1}$ optimális megoldása kielégíti a (2.4)—(2.6) egyenletrendszert. Részletezve: (2.2) alapján, felhasználva a $g_{j+1} > 0, g_{j+2} > 0, \ldots. g_k > 0$ relációkat:

$$\begin{aligned} u_{j+1} + u_{n+1} &= -c_{j+1}, \\ u_{j+2} + u_{n+1} &= -c_{j+2}, \\ &\vdots \\ u_k \quad + u_{n+1} &= -c_k. \end{aligned} \tag{2.4}$$

(2.3)-t és $g_{k+1}>0, g_{k+2}>0, \ldots, g_n>0$ relációkat felhasználva:

$$\begin{aligned} u_{k+1} &= 0, \\ u_{k+2} &= 0, \\ &\vdots \\ u_n &= 0. \end{aligned} \tag{2.5}$$

És természetesen eleget kell tenniök az

$$u_i \geqq 0, \qquad i = 1, \ldots, n, \tag{2.6}$$

$$u_i + u_{n+1} \geqq -c_i, \quad i = 1, \ldots, n$$

egyenlőtlenségeknek.

(2.4)-t átalakítjuk oly módon, hogy az u_{n+1} változót az egyenletrendszer paraméterének tekintjük:

$$\begin{aligned} u_{j+1} &= -c_{j+1} - u_{n+1}, \\ u_{j+2} &= -c_{j+2} - u_{n+1}, \\ &\vdots \\ u_k &= -c_k - u_{n+1}. \end{aligned}$$

Minthogy $u_i \geqq 0$ $(i=1, 2, \ldots, n)$:

$$\begin{aligned} u_{n+1} &\leqq -c_{j+1}, \\ u_{n+1} &\leqq -c_{j+2}, \\ &\vdots \\ u_{n+1} &\leqq -c_k. \end{aligned} \tag{2.7}$$

A (2.5) és (2.6) összeolvasztásából azt kapjuk, hogy

$$\begin{aligned} u_{n+1} &\geqq -c_{k+1}, \\ u_{n+1} &\geqq -c_{k+2}, \\ &\vdots \\ u_{n+1} &\geqq -c_n. \end{aligned} \tag{2.8}$$

A (2.7) és (2.8) egyenlőtlenségeket összefoglaljuk:

$$\max(-c_{k+1}, -c_{k+2}, \ldots, -c_n) \leqq u_{n+1} \leqq \min(-c_{j+1}, -c_{j+2}, \ldots, -c_k),$$

azaz

$$-c_{k+1} \leqq u_{n+1} \leqq -c_k.$$

Minthogy az (1.5) feladatnak olyan optimális megoldását keressük, amely extremális pontja a feladat megengedett tartományának, ezért szükségképpen csak a

$$4/a \textit{ eset}: \quad u_{n+1} = -c_k,$$

$$4/b \textit{ eset}: \quad u_{n+1} = -c_{k+1}$$

választások jöhetnek szóba. Ennek alapján a *4/a esetben*

$$\begin{aligned} u_{j+1} &= c_k - c_{j+1}, \\ u_{j+2} &= c_k - c_{j+2}, \\ &\vdots \\ u_{k-1} &= c_k - c_{k-1}, \\ u_k &= 0. \end{aligned}$$

Meghatározandók még az $u_1, u_2, \ldots, u_j$ változók értékei! (2.6) és $u_{n+1} = -c_k$ alapján eleget kell tenniök az

$$\begin{gathered} u_1 \geqq c_k - c_1, \\ u_2 \geqq c_k - c_2 \\ \vdots \\ u_j \geqq c_k - c_j, \\ u_1 \geqq 0,\ u_2 \geqq 0, \ldots, u_j \geqq 0 \end{gathered}$$

egyenlőtlenségeknek. Minthogy a megengedett tartomány csúcsát keressük:

$$\begin{aligned} u_1 &= \max(c_k - c_1, 0), \\ u_2 &= \max(c_k - c_2, 0), \\ &\vdots \\ u_j &= \max(c_k - c_j, 0). \end{aligned}$$

Tehát a *4/a esetben* az (1.5) feladat keresett optimális megoldása:

$$\begin{aligned} u_1 &= \max(c_k - c_1, 0), \\ u_2 &= \max(c_k - c_2, 0), \\ &\vdots \\ u_j &= \max(c_k - c_j, 0), \\ u_{j+1} &= c_k - c_{j+1}, \\ &\\ u_{j+2} &= c_k - c_{j+2}, \\ &\vdots \\ u_{k-1} &= c_k - c_{k-1}, \\ u_k &= 0, \\ u_{k+1} &= 0, \\ &\vdots \\ u_n &= 0, \\ u_{n+1} &= -c_k. \end{aligned}$$

A *4/b esetben* a 4/*a* esetben követett gondolatmenetet alkalmazzuk, és így az (1.5)

feladat következő megoldásához jutunk:

$$\begin{aligned}
u_1 &= \max(c_{k+1}-c_1, 0),\\
u_2 &= \max(c_{k+1}-c_2, 0),\\
&\vdots\\
u_j &= \max(c_{k+1}-c_j, 0),\\
u_{j+1} &= c_{k+1}-c_{j+1},\\
u_{j+2} &= c_{k+1}-c_{j+2},\\
&\vdots\\
u_k &= c_{k+1}-c_k,\\
u_{k+1} &= 0,\\
u_{k+2} &= 0,\\
&\vdots\\
u_n &= 0,\\
u_{n+1} &= -c_{k+1}.
\end{aligned}$$

5. eset

$$g_1 = g_2 = \ldots = g_j = 0; \quad g_{j+1} > 0,\ g_{j+2} > 0, \ldots, g_n > 0;$$

$$g_1+g_2+\ldots+g_{k-1} < g_{n+1} < g_1+g_2+\ldots+g_k \quad (j < k \leqq n).$$

Ekkor az (1.4) feladat optimális megoldása, melyet a c_j együtthatók rendezésével a mohó algoritmus szolgáltat, a következő:

$$x_1 = x_2 = \ldots = x_j = 0, \tag{2.9}$$

$$\begin{aligned}
x_{j+1} &= g_{j+1},\\
x_{j+2} &= g_{j+2},\\
&\vdots\\
x_{k-1} &= g_{k-1},\\
x_k &= g_{n+1}-g_{j+1}-g_{j+2}-\ldots-g_{k-1} < g_k,
\end{aligned} \tag{2.10}$$

$$\begin{aligned}
x_{k+1} &= 0,\\
x_{k+2} &= 0,\\
&\vdots\\
x_n &= 0.
\end{aligned} \tag{2.11}$$

Ismét a kiegészítő eltérések tételének alkalmazásával a (2.12)—(2.14) egyenletrendszert kapjuk az u_i változókra. Minthogy $x_{j+1}>0$, $x_{j+2}>0$, ..., $x_k>0$:

$$\begin{aligned}
u_{j+1}+u_{n+1} &= -c_{j+1},\\
u_{j+2}+u_{n+1} &= -c_{j+2},\\
&\vdots\\
u_k+u_{n+1} &= -c_k.
\end{aligned} \tag{2.12}$$

Mivel $g_{k+1}>0, g_{k+2}>0, \ldots, g_n>0$, az (1.4) feladat $k+1$-edik, $k+2$-edik, ..., n-edik feltételében a $\prec$ reláció teljesül, és minthogy $x_k<g_k$, ezért:

$$\begin{aligned} u_k &= 0, \\ u_{k+1} &= 0, \\ &\vdots \\ u_n &= 0, \end{aligned} \tag{2.13}$$

és természetesen most is fenn kell állnia a következő egyenlőtlenségeknek:

$$\begin{aligned} u_i &\geqq 0, \qquad i = 1, 2, \ldots, n, \\ u_i + u_{n+1} &\geqq -c_i, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{2.14}$$

A (2.12), (2.13) egyenletrendszerből: $u_k=0$, ezért $u_{n+1}=-c_k$, így:

$$\begin{aligned} u_{j+1} &= c_k - c_{j+1} \\ u_{j+2} &= c_k - c_{j+2} \\ &\vdots \\ u_{k-1} &= c_k - c_{k-1}, \end{aligned}$$

és (2.14)-ből a 4. esethez hasonló gondolatmenettel nyerhetjük az $u_1, u_2, \ldots, u_j$ értékeit. Összefoglalva, az (1.5) feladat keresett optimális megoldása:

$$\begin{aligned} u_1 &= \max(c_k - c_1, 0), \\ u_2 &= \max(c_k - c_2, 0), \\ &\vdots \\ u_j &= \max(c_k - c_j, 0), \\ u_{j+1} &= c_k - c_{j+1}, \\ u_{j+2} &= c_k - c_{j+2}, \\ &\vdots \\ u_{k-1} &= c_k - c_{k-1}, \\ u_k &= 0, \\ u_{k+1} &= 0, \\ &\vdots \\ u_n &= 0, \\ u_{n+1} &= -c_k. \end{aligned}$$

6. eset

$$g_1 \geqq 0, \; g_2 \geqq 0, \ldots, g_n \geqq 0; \quad g_1 + g_2 + \ldots + g_n \prec g_{n+1}.$$

Ekkor az (1.4) feladatnak nincs megengedett megoldása. Lévén az (1.5) feladatnak mindig van megengedett megoldása, a dualitás tétel értelmében most a célfüggvény nem korlátos. Némi számolással — melyet az olvasóra bízunk — belátható, hogy az

$$\mathbf{e}_1 + \mathbf{e}_2 + \ldots + \mathbf{e}_n - \mathbf{e}_{n+1}$$

vektor az (1.5) feladat negatív célfüggvényértékű extremális iránya.

3. Az (1.5) feladat megoldásának algoritmusa (összefoglalás)

1. lépés: Vizsgáljuk meg a $g_1, g_2, \ldots, g_{n+1}$ együtthatókat. Ha valamennyi nemnegatív, térjünk rá a 2. lépésre. Ha például $g_i<0$, akkor $\mathbf{e}_i$ negatív célfüggvényértékű extremális irány. *Itt az eljárás befejeződik.*

2. lépés: Ha $g_{n+1}>0$ térjünk rá a 3. lépésre.
Ha $g_{n+1}=0$, akkor számítsuk ki a

$$c = \max(-c_1, -c_2, \ldots, -c_n)$$

értéket. Ekkor $c \cdot \mathbf{e}_{n+1}$ optimális extremális pont. *Itt az eljárás befejeződik.*

3. lépés: Ha $g_1+g_2+\ldots+g_n \geqq g_{n+1}$, akkor térjünk rá a 4. lépésre.
Ha $g_1+g_2+\ldots+g_n<g_{n+1}$, akkor az

$$\mathbf{e}_1+\mathbf{e}_2+\ldots+\mathbf{e}_n-\mathbf{e}_{n+1}$$

negatív célfüggvényértékű extremális irány. *Itt az eljárás befejeződik.*

4. lépés: Oldjuk meg az (1.4) feladatot a mohó algoritmus segítségével. Legyen

$$J = \{j:\ 1 \leqq j \leqq n,\ g_j = 0\}.$$

A J halmazbeli indexeket kivéve rendezzük a változókat növekvő célfüggvényegyütthatóik szerint sorba. Adjuk meg az (1.4) feladat $x_1, x_2, \ldots, x_n$ megoldását. Legyen K a nem nulla szinten levő változók halmaza, és k az az index ezek közül, amelyhez tartozó célfüggvényegyüttható a legnagyobb. Legyen l a célfüggvényegyütthatók rendezésében a c_k-t követő együttható indexe.
Ha $x_k=g_k$ és $c_k \neq c_l$, akkor két optimális extremális pontot is nyerhetünk (a $c_k=c_l$ esetben a két megoldás azonos):

a) $u_{n+1} = -c_k$

$u_j = \max(c_k-c_j, 0)$, ha $j \in J$,

$u_j = c_k-c_j$, ha $j \in K$,

$u_j = 0$, egyébként.

b) $u_{n+1} = -c_l$,

$u_j = \max(c_l-c_j, 0)$, ha $j \in J$,

$u_j = c_l-c_j$, ha $j \in K$,

$u_j = 0$, egyébként.

Ha $x_k<g_k$, akkor a megoldást az a) pontban leírt értékek adják meg. *Itt az eljárás befejeződik.*

IRODALOM

[1] BENDERS, J. F., "Partitioning procedures for solving mixed variables programming problems", *Numerische Mathematik* 4 (1962) 238—252.
[2] DEÁK, I., HOFFER, J., MAYER, J., NÉMETH, Á., POTECZ, B., PRÉKOPA, A., STRAZICKY, B., "A short description of the optimal daily scheduling of the electricity production in Hungary", Working paper, MTA SZTAKI, 1980.
[3] HOFFER, J., „Megengedett megoldások vizsgálatával bővített Benders dekompozíciós algoritmus", *Alkalmazott Matematikai Lapok* **5** (1979) 177—190.
[4] KOVÁCS, L. B., *A diszkrét programozás kombinatorikus módszerei* (Bolyai János Matematikai Társulat, Budapest, 1969).
[5] PRÉKOPA, A., *Lineáris programozás I.* (Bolyai János Matematikai Társulat, Budapest, 1968).

(Beérkezett: 1981. március 14.)

HOFFER JÁNOS
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE
1502 BUDAPEST, XI. KENDE U. 13—17.

RÉSOLUTION DE PROBLÈMES D'APPROVISIONNEMENT ET DE PRODUCTION, À L'AIDE DE LA MÉTHODE DE BENDERS

J. HOFFER

Le problème à résoudre peut s'énoncer de la manière suivante: déterminer la valeur productive chacun des postes pour le même produit, dans le cas où les capacités des usines et le volume total à produire dépendent de décisions.

Utilisant la méthode de Benders: dans chaque itération on doit résoudre un sous-problème linéaire, dont le dual est un "problème du sac à dos". Adaptant "l'algorithme glouton" au dual, le théorème de la complémentarité permet de déterminer les points extrêmes et les directions d'infinitude du sous-problème original.

Alkalmazott Matematikai Lapok 7 (1981) 83—97

HIBRID OPTIMALIZÁLÁSI ELJÁRÁSOK NEM-DIFFERENCIÁLHATÓ SZTOCHASZTIKUS FELADATOK MEGOLDÁSÁRA

PINTÉR JÁNOS

Budapest

A dolgozatban sztochasztikusan kombinált optimalizálási eljárások egy általános osztályát vizsgáljuk, a leírt típusú módszerek lokális és globális konvergenciatulajdonságait bizonyítjuk. A vizsgált eljárások alkalmazhatók pl. folytonos, de nem szükségképpen differenciálható, véletlen zajjal torzított függvényekkel képezett matematikai programozási feladatok megoldására.

1. Bevezetés

A dolgozatban iteratív sztochasztikus optimalizálási eljárások egy általános osztályát vizsgáljuk, amelyek alkalmazhatók véletlen zajjal torzított struktúrájú feladatok megoldására. Például a sztochasztikus programozás egy általános feladattípusa a

$$\min_{\mathbf{x}\in K} M_{\mathbf{y}}[f(\mathbf{x}, \mathbf{y})],$$

$$P\{g_i(\mathbf{x}, \mathbf{y}) \geqq 0,\ i = 1, \ldots, m\} \geqq p, \tag{1.1}$$

alakban írható fel: itt $\mathbf{x}\in E^n$ — optimalizálandó vektor, az n-dimenziós euklideszi tér eleme; $\mathbf{y}\in E^q$ — vektorértékű valószínűségi változó; f, g_i, $i=1, \ldots, m$ — adott függvények; M_y — a várható érték képzésének operátora; $P\{.\}$ a $\{.\}$ véletlen esemény bekövetkezésének valószínűsége; $0<p<1$ rögzített paraméter; $K\subset E^n$ — korlátos, zárt halmaz.

Ismeretes, hogy az (1.1) feladatnak viszonylag enyhe feltételek mellett létezik globálisan optimális megoldása (pl. a megengedett halmaz nem-üres, korlátos, zárt volta és a célfüggvény folytonossága esetén, l. [30] II. fej. 8. §. 2.; egy a megoldás létezésére vonatkozó egyszerű állítást a dolgozat függelékében bizonyítunk). Ugyanakkor a feladat célfüggvénye és megengedett megoldásainak halmaza az optimalizálás során általában csak közelítőleg ismert (pl. az iteráció során szimulációval becsült függvényértékek alapján).

Determinisztikus (nemlineáris) és sztochasztikus programozási feladatok megoldására ismeretesek különböző iteratív optimalizálási eljárások, l. pl. [1, 7, 13, 25—29, 31, 32]. Ilyen módszerek, ill. alkalmazásaik magyar nyelvű leírását tartalmazzák pl. a [4, 8—10, 12, 15—20, 24] dolgozatok. Az említett módszerek alkalmazásakor a legtöbb esetben feltételezik, ill. igazolják a feladat függvényeinek különböző differenciálhatósági és/vagy konvexitási tulajdonságait, sztochasztikus feladat esetén a véletlen zajhatások iterációs lépésenkénti függetlenségét és centrált voltát. Emellett az optimalizálási eljárások irányválasztása gradiens típusú módszereken,

ill. ezek bizonyos általánosításain, lényegileg egyszerű véletlen keresésen vagy heurisztikus jellegű stratégiákon alapszik. A következőkben megvizsgáljuk, hogyan lehet a feladat szerkezetére vonatkozó követelményeket enyhíteni, másrészt, milyen feltételek mellett lehet más — esetleg effektívebb — irányválasztási stratégiákat alkalmazni.

Ismeretes pl., hogy az egyszerű véletlen keresésen alapuló optimalizálási módszereknek (l. pl. [33]) a feladat megoldásához való (1 valószínűségű) konvergenciájához általában elegendő az (1.1) feladatban szereplő függvények (x szerinti) folytonossága. (Ilyen típusú konvergenciaeredmény a jelen dolgozat 2. és 3. részében leírt tételekből is adódik egyszerű speciális esetként.) Megjegyezzük továbbá, hogy pl. feltétel nélküli determinisztikus feladatok megoldása esetén a gradiens irányú lépéseken alapuló módszerek már kvadratikus célfüggvényre sem végesek, jóllehet itt léteznek véges (konjugált gradiens típusú) eljárások [5, 14]. Ezek különböző variánsai, pl. [3, 23] gyakran hatékonyak maradnak általánosabb nemlineáris függvényosztályon történő optimalizálás esetén is. Példaként a [6] dolgozat részletes számítási eredményeit említjük, amelyek szerint *Powell módszere* kiemelkedően hatékony a gradiensmentes optimalizálási eljárások között (az eredmények egy részét korábbi dolgozatunkban idéztük, l. [10], 9. táblázat). Ezért várható, hogy pl. a konjugált gradiens típusú módszerek alkalmas variánsai lokálisan megfelelően hatékonyak maradnak sztochasztikus optimalizálási problémák megoldása esetén is, ha a véletlen ingadozásokkal, hibákkal becsülhető szerkezetű feladat „egyre pontosabban közelít" megfelelő simasági tulajdonságokkal rendelkező struktúrájú determinisztikus feladatot.

A fenti észrevételeket felhasználva lehetővé válik olyan optimalizálási eljárások létrehozása, amelyek folytonos (de nem szükségképpen differenciálható és/vagy konvex) feladatok megoldására is alkalmazhatók, emellett várhatóan lokálisan hatékonyak. Ilyen eljárások megadhatók pl. véletlen kereső és alkalmas lokális módszerek sztochasztikus kombinációján alapuló hibrid eljárásokként: a dolgozatban ezek vizsgálatával foglalkozunk. Megjegyezzük, hogy optimalizálási algoritmusokat heurisztikusan kombináló módszereket korábban is alkalmaztak. (Ezzel kapcsolatban pl. a [2] cikk megfelelő hivatkozásait említjük, hasonló heurisztikus módszert alkalmaztunk a [11] dolgozatban egy összetett vízgazdálkodási tervezési probléma megoldására.)

A dolgozat következő két pontjában sztochasztikus hibrid optimalizálási eljárások lokális, ill. globális konvergenciatulajdonságait vizsgáljuk. A lokális konvergenciatételek a [10] dolgozat 2.2. tételének, a globális konvergenciaeredmények pedig a [2] dolgozat 1. és 3. tételének megfelelő általánosításain alapszanak.

2. Lokális konvergenciatulajdonságok

Vizsgáljuk először a következő feltétel nélküli minimalizálási feladatot:

$$\min_{\mathbf{x}\in E^n} f(\mathbf{x}), \qquad f(\mathbf{x}) = F(\mathbf{x}, \xi), \tag{2.1}$$

ahol az $F(\mathbf{x}, \xi)$ függvény értéke az $\mathbf{x}$ döntési változón kívül a $\xi = \xi(\mathbf{x})$-szel jelölt véletlen ingadozásoktól is függ. Tegyük fel, hogy ez a függés additív, vagyis

$$F(\mathbf{x}, \xi) \approx f(\mathbf{x}) + \xi(\mathbf{x}), \tag{2.2}$$

és teljesülnek az alábbi 1—2. feltételcsoportok:

1. Az $F(\mathbf{x}, \xi)$ függvény $f(\mathbf{x})$ determinisztikus komponensére, vonatkozóan megköveteljük, hogy

a) $f(\mathbf{x})$ folytonos függvény, amelynek minimumhelyei sehol sem torlódnak;
b) $\lim_{\|\mathbf{x}\| \to \infty} f(\mathbf{x}) = \infty$ ($\|\mathbf{x}\|$ az $\mathbf{x}$ vektor euklideszi normája);
c) ha $\hat{\mathbf{x}} \in E^n$ az f függvénynek nem lokális minimumhelye, akkor léteznek olyan $\hat{\delta}(\hat{\mathbf{x}}) > 0$ és $\hat{a}(\hat{\mathbf{x}}) \in (0, 1)$ számok, valamint $\hat{\mathbf{d}}(\hat{\mathbf{x}}) \in E^n$ egységvektor, amelyekkel minden $\mathbf{x} = \hat{\mathbf{x}} + \delta \mathbf{d}$ vektorra $0 < \delta \leqq \hat{\delta}$ és $\mathbf{d} \in E^n$ $\|\mathbf{d}\| = 1$, $(\mathbf{d}, \hat{\mathbf{d}}) \geqq 1 - \hat{a}$ esetén teljesül $f(\mathbf{x}) < f(\hat{\mathbf{x}})$.

Egyszerűen belátható, hogy az 1/*c*. feltételt teljesíti minden folytonosan differenciálható függvény. A [10] dolgozatban mutattunk példát olyan folytonos függvényre, amelyik ezt a feltételt nem teljesíti.

2. Az $F(\mathbf{x}, \xi)$ függvény $\xi(\mathbf{x})$ sztochasztikus (zaj) komponensére vonatkozóan megköveteljük, hogy

a) $\xi(\mathbf{x}) = \xi(\mathbf{x}, \omega)$ az $\mathbf{x} \in E^n$ paramétertől függő valószínűségi változó (vagyis minden $\mathbf{x} \in E^n$ esetén $\xi(\mathbf{x}, \omega)$: $\Omega \to E^1$ ($\omega \in \Omega$) valamely (Ω, $\mathcal{A}$, P) valószínűségi mezőn értelmezett *P-mérhető függvény*);
b) létezik olyan $0 < S < \infty$ állandó, hogy a következőkben leírt típusú iteratív hibrid eljárás által generált bármely $\{\mathbf{x}_k\}$ vektorsorozat esetén teljesül

$$\left| \sum_{k=1}^{\infty} \xi(\mathbf{x}_k, \omega) \right| \leqq S$$

1 valószínűséggel (majdnem mindenütt, ezt a továbbiakban a szokásos (m.m.) szimbólummal fogjuk jelölni).

Kiemeljük, hogy mind a 2/*b*. feltételben, mind pedig a dolgozat további hasonló feltételeiben ezek ω szerint egyenletes teljesülését írjuk elő.

3. A (2.1) feladat megoldására a következő eljárástípust alkalmazzuk. Legyen $\mathbf{x}_0 \in E^n$ a megoldás tetszőleges kezdeti közelítése. A $(k+1)$-edik pont $k = 0, 1, 2, \ldots$ esetén az

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma_k \mathbf{s}_k \tag{2.3}$$

iterációval adódik. Az $\mathbf{s}_k \in E^n$, $\|\mathbf{s}_k\| = 1$ továbbhaladási irány és a $\gamma_k \geqq 0$ lépéshossz megválasztása a következő stratégia szerint történik.

a) Az irányválasztást minden lépésben az

$$\mathbf{s}_k\colon \beta_k \mathbf{R}_k + (1 - \beta_k)\mathbf{L}_k \quad 0 \leqq \beta_k \leqq 1 \tag{2.4}$$

szimbolikus előírás szerint végezzük. Ez azt jelenti, hogy β_k valószínűséggel az n-dimenziós egységgömb felületén (a korábbiaktól független) egyenletes eloszlású véletlen vektor generálásával választunk irányt ($\mathbf{R}_k = \mathbf{R}$ eljárás $k = 0, 1, 2, \ldots$), $1 - \beta_k$ valószínűséggel pedig valamely más $\mathbf{L}_k$ lokális optimalizáló eljárás szerint. Emellett feltesszük, hogy létezik olyan $\beta > 0$ szám, amellyel teljesül $\beta_k \geqq \beta$, $k = 0, 1, 2, \ldots$.

b) Az $\mathbf{s}_k$ irány kiválasztása után γ_k közelítő lokális iránymenti minimalizálással adódik. Ennek eredményeként előírjuk, hogy (m.m.) teljesüljön az

$$F(\mathbf{x}_{k+1}, \xi) \leqq \min_{0 \leqq \gamma \leqq h_k} f(\mathbf{x}_k + \gamma \mathbf{s}_k) + \varepsilon_{k+1} \tag{2.5}$$

egyenlőtlenség. Itt $h_k \geqq h > 0$ (h állandó), az $\varepsilon_k = \varepsilon_k(\mathbf{x}_k, \omega) \geqq 0$ maximális véletlen hibákra vonatkozóan pedig előírjuk a

$$\sum_{k=1}^{\infty} \varepsilon_k \leqq E$$

(m.m.) feltételt ($0 < E < \infty$ az $\{\mathbf{x}_k\}$ sorozat kiválasztásától független állandó).

A (2.5) relációval kapcsolatban megjegyezzük, hogy a 2/b. feltétel következtében ez elvileg könnyen teljesíthető pl. *Lipschitz-folytonos f függvény* esetén (az $[\mathbf{x}_k, \mathbf{x}_k + h_k \mathbf{s}_k]$ szakaszon felvett alkalmasan sűrű rácson való optimalizálással).

A fenti 1—3. feltételekből látható, hogy a (2.1) feladat iteratív megoldásának egy modelljéről van szó, amelyben az iteráció folyamán a célfüggvény kiértékelése és a lokális iránymenti optimalizálások végrehajtása egyaránt fokozatosan pontosabbá válik. (A gyakorlatban sztochasztikus zajjal torzított függvények egyre pontosabb kiértékelése pl. a függvény értékét befolyásoló véletlen tényezők növekvő számú realizációjával — kísérletek, számítógépi szimuláció — biztosítható.) Emellett az irányválasztási stratégiára vonatkozóan csupán azt írjuk elő, hogy abban a véletlen irányválasztás lehetősége a 3/*a*. feltétel szerint „kellő súllyal" szerepeljen.

2.1. TÉTEL. Az 1—3. feltételek fennállása esetén a generált $\{\mathbf{x}_k\}$ vektorsorozat korlátos (m.m.) és bármely konvergens részsorozata az $f(\mathbf{x})$ függvény valamely lokális minimumhelyéhez konvergál (m.m.).

Bizonyítás. Először belátjuk, hogy $\{\mathbf{x}_k\}$ korlátos (m.m.). Legyen $\mu > 0$ tetszőleges érték. A

$$\sum_{j=1}^{\infty} \xi(\mathbf{x}_j)$$

sor konvergenciája következtében létezik olyan $k(\mu)$ index, hogy $k \geqq k(\mu)$ esetén

$$\left| \sum_{j=k+2}^{\infty} \xi(\mathbf{x}_j) \right| \leqq \mu.$$

Ezért $k \geqq k(\mu)$ esetén (2.5) alapján érvényesek az alábbi egyenlőtlenségek (m.m.):

$$\begin{aligned} f(\mathbf{x}_{k+1}) &\leqq \min_{0 \leqq \gamma \leqq h_k} f(\mathbf{x}_k + \gamma \mathbf{s}_k) + \varepsilon_{k+1} - \xi(\mathbf{x}_{k+1}) \leqq f(\mathbf{x}_k) + \varepsilon_{k+1} - \xi(\mathbf{x}_{k+1}) \leqq \\ &\leqq \ldots \leqq f(\mathbf{x}_0) + \sum_{j=1}^{k+1} \varepsilon_j - \sum_{j=1}^{k+1} \xi(\mathbf{x}_j) \leqq f(\mathbf{x}_0) + E + \left| \sum_{j=1}^{\infty} \xi(\mathbf{x}_j) \right| + \\ &+ \left| \sum_{j=k+2}^{\infty} \xi(\mathbf{x}_j) \right| \leqq f(\mathbf{x}_0) + E + S + \mu. \end{aligned} \tag{2.6}$$

Az $\{f(\mathbf{x}_k)\}_{k\geqq k(\mu)}$ számsorozat korlátosságából és a tétel 1/*b*. feltételéből következik, hogy az $\{\mathbf{x}_k\}$ sorozat korlátos (m.m.). Az egyszerűség kedvéért jelöljük $\{\mathbf{x}_k\}$-val ennek egy konvergens részsorozatát és legyen $\lim_{k\to\infty}\mathbf{x}_k=\hat{\mathbf{x}}$: megmutatjuk, hogy $\hat{\mathbf{x}}$ lokális minimumhelye $f(\mathbf{x})$-nek (m.m.).

Tegyük fel, hogy ez nem igaz, akkor az 1/*c*. feltétel értelmében létezik $\hat{\delta}>0$, $0<\hat{a}<1$ és $\hat{\mathbf{d}}\in E^n\ \|\hat{\mathbf{d}}\|=1$, amelyekkel bármely $\mathbf{x}=\hat{\mathbf{x}}+\delta\mathbf{d}$ vektorra $0<\delta\leqq\hat{\delta}$, $\mathbf{d}\in E^n\ \|\mathbf{d}\|=1$, $(\mathbf{d},\hat{\mathbf{d}})\geqq 1-\hat{a}$ esetén teljesül $f(\mathbf{x})<f(\hat{\mathbf{x}})$.

Jelölje $K(\hat{\mathbf{x}},\hat{a},\hat{\delta},\hat{\mathbf{d}})$ ezen $\mathbf{x}$ pontok halmazát, legyen továbbá $G(\hat{\mathbf{x}},\hat{\delta})$ az $\hat{\mathbf{x}}$ középpontú, $\hat{\delta}$ sugarú n-dimenziós gömb: itt $\hat{\delta}$ értékét a $\hat{\delta}<\frac{h}{2}$ egyenlőtlenségnek megfelelően választjuk meg. Mivel $\lim_{k\to\infty}\mathbf{x}_k=\hat{\mathbf{x}}$, bármely $\hat{\delta}>0$ esetén teljesül $\mathbf{x}_k\in G(\hat{\mathbf{x}},\hat{\delta})$, ha $k\geqq k(\hat{\delta})$. Igazoljuk, hogy minden $k_0\geqq k(\hat{\delta})$ esetén az $\{\mathbf{x}_k\}_{k\geqq k_0}$ részsorozat (m.m.) tartalmaz olyan pontot, amelyben $f(\mathbf{x}_k)<f(\hat{\mathbf{x}})-v$ (itt $v>0$ az alábbiakban definiált érték).

Vezessük be az $\bar{\mathbf{x}}=\underset{\mathbf{x}\in K(\hat{\mathbf{x}},\hat{a},\hat{\delta},\hat{\mathbf{d}})}{\arg\min} f(\mathbf{x})$ jelölést. Az $f(\mathbf{x})$ folytonossága miatt létezik olyan $\bar{\delta}>0$ sugarú $G(\bar{\mathbf{x}},\bar{\delta})$ gömb, hogy $\mathbf{x}\in G(\bar{\mathbf{x}},\bar{\delta})$ esetén teljesül $f(\mathbf{x})\leqq f(\hat{\mathbf{x}})-v$ (pl. $v=f(\hat{\mathbf{x}})-\max_{\mathbf{x}\in G(\bar{\mathbf{x}},\bar{\delta})} f(\mathbf{x})>0$ definícióval).

Definiáljuk az A_{k_0} $k_0\geqq k(\hat{\delta})$ és $\{B_k\}_{k\geqq k_0}$ véletlen eseményeket a következő módon:

A_{k_0} — a korábban leírt sztochasztikus hibrid eljárás a $k\geqq k_0$ indexű iterációs lépések valamelyikében kiválaszt olyan továbbhaladási irányt, amelyik tartalmaz az $f(\mathbf{x})\leqq f(\hat{\mathbf{x}})-v$ relációt kielégítő $\mathbf{x}$ pontot;

B_k — az eljárás a k-adik lépésben vizsgálja az $N(\hat{\mathbf{x}},\bar{\mathbf{x}})\overset{\text{def}}{=}K(\hat{\mathbf{x}},\hat{a},\hat{\delta},\hat{\mathbf{d}})\cap G(\bar{\mathbf{x}},\bar{\delta})$ halmaz valamely pontját.

Nyilvánvaló, hogy $\bigcup_{k\geqq k_0} B_k\subseteq A_{k_0}$. A B_k esemény ellentettjét $\bar{B}_k$-sal jelölve az eljárás konstrukciója folytán teljesül az alábbi egyenlőtlenség:

$$P(B_k|\bar{B}_{k-1},\ldots,\bar{B}_{k_0}) = \tag{2.7}$$

$$= P(B_k|\bar{B}_{k-1},\ldots,\bar{B}_{k_0};\mathbf{R}_k)\cdot\beta_k+P(B_k|\bar{B}_{k-1},\ldots,\bar{B}_{k_0};\mathbf{L}_k)\cdot(1-\beta_k)\geqq p\cdot\beta_k.$$

Itt p a $G(\hat{\mathbf{x}},\hat{\delta})$ gömb pontjaiban történő véletlen irányválasztással az $N(\hat{\mathbf{x}},\bar{\mathbf{x}})$ halmazt metsző irány kiválasztási valószínűségének alsó határa; a (2.4) formulával kapcsolatban definiált $\mathbf{R}_k=\mathbf{R}$ irányválasztási stratégia és az $N(\hat{\mathbf{x}},\bar{\mathbf{x}})$ halmaz pozitív mértékű volta következtében p pozitív. A (2.7) relációból egyszerűen következik $P(A_{k_0})\geqq P(\bigcup_{k\geqq k_0} B_k)=1$, ui.

$$P\left(\bigcap_{k=k_0}^{n}\bar{B}_k\right)\leqq\prod_{k=k_0}^{n}(1-p\cdot\beta_k)\leqq(1-p\beta)^{n-k_0+1}\to 0,\quad\text{ha}\quad n\to\infty,$$

tehát bármely $k_0\geqq k(\hat{\delta})$ esetén az $\{\mathbf{x}_k\}_{k\geqq k_0}$ részsorozat valamely pontjában az eljárás olyan irányt választ ki, amely tartalmaz az $f(\mathbf{x})\leqq f(\hat{\mathbf{x}})-v$ egyenlőtlenséget kielégítő $\mathbf{x}$ pontot (m.m.).

Vegyük figyelembe még, hogy az adott $\nu>0$ számhoz megadható olyan $k(\nu)$ index, hogy $k\geqq k(\nu)$ esetén $|\xi(\mathbf{x}_k)|<\frac{\nu}{4}$ (m.m.) és $\varepsilon_k<\frac{\nu}{4}$ (m.m.) is teljesül, ezért ha $k_0\geqq\max(k(\nu), k(\hat{\delta}))$, akkor az $\{\mathbf{x}_k\}_{k\geqq k_0}$ részsorozat tartalmaz olyan pontot (legyen ez $\mathbf{x}_{k_1}$), amelyre teljesül $f(\mathbf{x}_{k_1})<f(\hat{\mathbf{x}})-\frac{\nu}{4}$ (l. a (2.5) feltételt). Ekkor viszont a további pontokban fennáll

$$f(\mathbf{x}_k)\leqq f(\mathbf{x}_{k_1})+\sum_{j=k_1+1}^{k}\varepsilon_j-\sum_{j=k_1+1}^{k}\xi(\mathbf{x}_j). \tag{2.8}$$

Ha most még feltesszük, hogy $k\geqq k_0$ esetén $\sum_{j=k}^{\infty}\varepsilon_j<\frac{\nu}{16}$ és $\left|\sum_{j=k}^{\infty}\xi(\mathbf{x}_j)\right|<\frac{\nu}{16}$ is teljesül (m.m.), akkor (2.8)-ból $k\geqq k_1\geqq k_0$ esetén adódik (m.m.)

$$f(\mathbf{x}_k)<f(\hat{\mathbf{x}})-\frac{\nu}{4}+\frac{\nu}{16}+\left|\sum_{j=k_1+1}^{k}\xi(\mathbf{x}_j)\right|\leqq f(\hat{\mathbf{x}})-\frac{3\nu}{16}+ \tag{2.9}$$

$$+\left|\sum_{j=k_1+1}^{\infty}\xi(\mathbf{x}_j)\right|+\left|\sum_{j=k+1}^{\infty}\xi(\mathbf{x}_j)\right|<f(\hat{\mathbf{x}})-\frac{\nu}{16}.$$

Ez viszont az $f(\mathbf{x})$ függvény folytonossága miatt ellentmond a $\lim_{k\to\infty}\mathbf{x}_k=\hat{\mathbf{x}}$ relációnak. Ezért $\hat{\mathbf{x}}$ valóban lokális minimumhely (m. m.).

A továbbiakban a 2.1 tételnek a

$$\min_{\mathbf{x}\in C} f(\mathbf{x})\quad f(\mathbf{x})\approx F(\mathbf{x},\xi),\quad C\approx C(\eta) \tag{2.10}$$

korlátozó feltételek melletti optimalizálási feladatra vonatkozó kiterjesztésével foglalkozunk. A (2.10) feladatban mind az $f(\mathbf{x})$ célfüggvény, mind pedig a C megengedett tartomány csak közelítőleg adott (azaz számítható) az iteráció során. A 2.1. tétel feltételeit a következőképpen módosítjuk, ill. egészítjük ki:

1. Az $f(\mathbf{x})$ determinisztikus komponensre vonatkozó *a)*, *b)*, *c)* feltételek érvényben maradnak, emellett előírjuk, hogy *d)* a 2.1. tétel bizonyításában szereplő $K(\hat{\mathbf{x}}, \hat{a}, \hat{\delta}, \hat{\mathbf{d}})$ halmazt (minden $\hat{\mathbf{x}}$ nem lokális minimumhely esetén) tartalmazza a $C(\eta)$ halmaz C determinisztikus komponense (l. később a $C(\eta)$ struktúráját leíró 4. feltételt).

2. A $\xi(\mathbf{x})$ sztochasztikus komponensre vonatkozó *a)*, *b)* feltételek érvényben maradnak.

3. Az alkalmazható sztochasztikus hibrid algoritmusok szerkezetére vonatkozó előírások közül *a)* változatlan, *b)* helyett pedig a $C(\eta)$ halmazon végrehajtandó közelítő lokális iránymenti optimalizálást írunk elő úgy, hogy az adódó $\mathbf{x}_{k+1}$ pontra $k=0, 1, 2, \ldots$ esetén teljesüljön (m.m.)

$$F(\mathbf{x}_{k+1},\xi)\leqq\min_{\substack{0\leqq\gamma\leqq h_k\\ \mathbf{x}_k+\gamma\mathbf{s}_k\in C(\eta)}} f(\mathbf{x}_k+\gamma\mathbf{s}_k)+\varepsilon_{k+1}. \tag{2.11}$$

Itt feltesszük még, hogy $\mathbf{x}_k \in C(\eta)$ $k=0, 1, 2, \ldots$ és (mint korábban) előírjuk a $h_k \geqq h > 0$ és $\sum_{k=1}^{\infty} \varepsilon_k \leqq E < \infty$ (m.m.) feltételeket.

4. A (2.10) feladatban szereplő, véletlen zajjal torzított megengedett tartományról feltesszük, hogy az iteráció k-adik lépésében ($k=0, 1, 2, \ldots$) érvényes a

$$C(\eta) = C + \eta(\mathbf{x}_k) \tag{2.12}$$

összeg alakú előállítás (vagyis minden $\mathbf{x} \in C(\eta)$ felírható $\mathbf{x}=\mathbf{y}+\mathbf{z}$ alakban, ahol $\mathbf{y} \in C$, $\mathbf{z} \in \eta(\mathbf{x}_k)$). Emellett

a) a $C(\eta)$ halmaz C determinisztikus komponense az E^n tér valamely nemüres, összefüggő, nyílt halmazának lezártja;

b) $\eta(\mathbf{x}) = \eta(\mathbf{x}, \omega)$ véletlen pont-halmaz leképezés, vagyis minden $\mathbf{x} \in E^n$ és $\omega \in \Omega$ esetén $\eta(\mathbf{x}, \omega)$ E^n valamely részhalmaza, és ez a leképezés *P-mérhető*. Feltesszük továbbá, hogy a $C(\eta)$ direkt összeg alakú halmaz C-hez hasonlóan nem-üres, összefüggő, nyílt halmaz lezártja E^n-ben úgy, hogy a fenti 3. *a)*, *b)* feltételekkel leírt optimalizálási eljárás végrehajtható legyen. Definiáljuk az $\eta(\mathbf{x}, \omega)$ halmaz normáját az $\|\eta(\mathbf{x}, \omega)\| = \text{vrai sup}\, \|\mathbf{z}\|$ $\mathbf{z} \in \eta(\mathbf{x}, \omega)$ előírással és tegyük fel, hogy teljesül (m.m.) $\sum_{k=0}^{\infty} \|\eta(\mathbf{x}_k, \omega)\| \leqq T$, ahol $0 < T < \infty$ a generált $\{\mathbf{x}_k\}$ sorozattól független állandó.

Megjegyezzük, hogy a 4/*a*, *b*. feltételek a $C(\eta)$ halmaz C determinisztikus komponense egyre pontosabb meghatározásának egy modelljét írják le. Érvényes a 2.1. tétellel analóg következő állítás:

2.2. TÉTEL. A felsorolt 1—4. feltételek fennállása esetén az $\{\mathbf{x}_k\}$ sorozat korlátos (m.m.) és bármely konvergens részsorozata az $f(\mathbf{x})$ függvénynek a C halmazbeli valamely lokális minimumhelyéhez tart (m.m.).

Bizonyítás. Ismét indirekt módon okoskodunk. (A bizonyítást a 2.1. tétel igazolásához hasonlóan végezzük, ezért csak az ott leírtaktól eltérő részleteket taglaljuk.) Az $\mathbf{x}_k \in C(\eta)$ feltétel miatt a korábbihoz teljesen analóg módon nyerhető itt is a (2.6) egyenlőtlenség-lánc, amelyből az $\{\mathbf{x}_k\}$ sorozat korlátos volta következik. Válasszunk ki $\{\mathbf{x}_k\}$-ból (változatlan jelölés mellett) egy konvergens részsorozatot és legyen $\lim_{k\to\infty} \mathbf{x}_k = \hat{\mathbf{x}}$. A (2.11) formula által leírt közelítő minimalizálási eljárás alapján $\mathbf{x}_{k+1} \in C + \eta(\mathbf{x}_k)$ $k=0, 1, 2, \ldots$, ezért a C halmaz zártsága és a $\lim_{k\to\infty} \|\eta(\mathbf{x}_k, \omega)\| = 0$ feltevés miatt $\hat{\mathbf{x}} \in C$ (m.m.). Ha most $\hat{\mathbf{x}}$ $f(\mathbf{x})$-nek nem lokális minimumhelye a C halmazon, akkor a 2.1. tétel bizonyításában leírtak analógiájára definiálható a $K(\hat{\mathbf{x}}, \hat{a}, \hat{\delta}, \hat{\mathbf{d}}) \subset C$ halmaz (itt $\hat{\delta} < \frac{h}{2}$), a $G(\bar{\mathbf{x}}, \delta)$ gömb és a $v > 0$ érték, amelyekkel az $N_k(\hat{\mathbf{x}}, \bar{\mathbf{x}}) \overset{\text{def}}{=} K(\hat{\mathbf{x}}, \hat{a}, \hat{\delta}, \hat{\mathbf{d}}) \cap G(\bar{\mathbf{x}}, \delta) \cap \{C + \eta(\mathbf{x}_k)\}$ halmaz pontjaiban fennáll $f(\mathbf{x}) \leqq f(\hat{\mathbf{x}}) - v$. A korábbiakhoz hasonlóan meggyőződhetünk arról, hogy elég nagy k_0 indextől kezdve az $\{\mathbf{x}_k\}_{k \geqq k_0}$ részsorozat valamely pontjában (m.m.) sor kerül olyan irány kiválasztására, amely áthalad a pozitív mértékű $\bigcap_{k \geqq k_0} N_k(\hat{\mathbf{x}}, \bar{\mathbf{x}})$ halmazon (itt kihasználjuk a $C(\eta)$ halmazok szerkezetére vonatkozó 4/*a*, *b*. feltételeket).

Ezért elég pontos célfüggvény-kiértékelések és elég kis iránymenti minimalizálási hibák esetén — éppúgy mint a 2.1. tétel bizonyításában — (m.m.) adódik olyan $\mathbf{x}_{k_1}$ $k_1 \geqq k_0$, amelyre $f(\mathbf{x}_{k_1}) < f(\hat{\mathbf{x}}) - \frac{v}{4}$. Ebből a (2.8) és (2.9) formulák felhasználásával adódik (m.m.) az $f(\mathbf{x}_k) < f(\hat{\mathbf{x}}) - v_1$ reláció ($v_1 > 0$ alkalmas érték), ha $k \geqq k_1 \geqq k_0$, tehát — $f(\mathbf{x})$ folytonosságát is figyelembevéve — $\{\mathbf{x}_k\}$ nem konvergálhat az $\hat{\mathbf{x}}$ ponthoz. A kapott ellentmondás mutatja, hogy $\hat{\mathbf{x}}$ lokális minimumhelye f-nek a C halmazon (m.m.).

3. Globális konvergenciatulajdonságok

A globális minimum meghatározására vonatkozó feladatot először determinisztikus alakban vizsgáljuk:

$$\min_{\mathbf{x} \in C} f(\mathbf{x}). \tag{3.1}$$

A feladat struktúráját és az alkalmazandó optimalizáló algoritmust illetően a következő feltevésekkel élünk:

1. $f(\mathbf{x})$ konstanstól különböző folytonos függvény a $C \subset E^n$ halmazon;
2. C valamely nem-üres, korlátos, összefüggő, nyílt halmaz lezártja (az 1., 2. feltételek értelmében (3.1)-nek létezik optimális megoldása);
3. A (3.1) feladat megoldására a következőkben leírt sztochasztikusan kombinált eljárástípust alkalmazzuk. Legyen $\mathbf{x}_0 \in C$ a globális optimumhely tetszőleges közelítése. Az $\{\mathbf{x}_k\}$ vektorsorozat további elemeit az

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma_k \mathbf{s}_k = \mathbf{H}_k(\mathbf{x}_1, \ldots, \mathbf{x}_k); \quad \mathbf{x}_k \in C \quad k = 0, 1, 2, \ldots \tag{3.2}$$

iterációval állítjuk elő, ahol $\mathbf{H}_k$ valamely

$$\mathbf{H}_k\colon \alpha_k \mathbf{G}_k + (1-\alpha_k)\mathbf{L}_k \quad 0 \leqq \alpha_k \leqq 1, \quad \sum_{k=1}^{\infty} \alpha_k = \infty \tag{3.3}$$

alakú hibrid eljárás. Ez azt jelenti, hogy az előző iterációk eredményétől függetlenül α_k valószínűséggel a $\mathbf{G}_k$ globális, véges lépésszámú szubalgoritmus (részeljárás), $(1-\alpha_k)$ valószínűséggel pedig valamely véges lépésszámú $\mathbf{L}_k$ lokális szubalgoritmus kerül végrehajtásra.

Legyen $f^* = f(\mathbf{x}^*) = \min_{\mathbf{x} \in C} f(\mathbf{x})$. A $\mathbf{G}_k$ részeljárásokról feltesszük, hogy bármely $\varepsilon > 0$ esetén $\mathbf{G}_k = \mathbf{G}_k(\varepsilon)$ véges számú lépésben lehetővé teszi az

$$X_\varepsilon = \{\mathbf{x}_\varepsilon \in C : f(\mathbf{x}_\varepsilon) \leqq f^* + \varepsilon\} \tag{3.4}$$

ε-optimális halmazba eső pont kiválasztását, alkalmas $0 < p_k(\varepsilon) \leqq 1$ értéknél nem kisebb valószínűséggel. Legyen továbbá $p_k(\varepsilon)$ $k = 0, 1, 2, \ldots$ esetén ε monoton növekvő függvénye és tegyük fel, hogy minden $\varepsilon > 0$ és minden, a (3.2)—(3.3) hibrid eljárás által generált $\{\mathbf{x}_k\}$ sorozat esetén teljesül

$$\liminf_{k \to \infty} p_k(\varepsilon) = p(\varepsilon) > 0. \tag{3.5}$$

(A leírt tulajdonságokkal rendelkező $\{\mathbf{G}_k\}$ szubalgoritmusok megválasztására a dolgozat 4. pontjában említünk példákat.)

Legyen $\varepsilon_k>0$, $\lim_{k\to\infty}\varepsilon_k=0$; a (3.2)—(3.3) eljárást egyre pontosabb $\mathbf{G}_k=\mathbf{G}_k(\varepsilon_k)$ globális részeljárások segítségével konstruáljuk meg. Definiáljuk az optimumnak és az optimális megoldásnak a k-adik iteráció utáni közelítését az

$$f_k = \min\{f(\mathbf{x}_0), \ldots, f(\mathbf{x}_k)\} \tag{3.6}$$
$$\bar{\mathbf{x}}_k = \arg\min\{f(\mathbf{x}_0), \ldots, f(\mathbf{x}_k)\}$$

formulákkal (ha egyszerre több $\bar{\mathbf{x}}_k$ létezik, akkor ezek közül a legkisebb indexűt választjuk ki). Érvényes a következő állítás:

3.1. TÉTEL. A (3.1) feladat és a sztochasztikusan kombinált optimalizálási eljárásokban alkalmazható $\mathbf{G}_k$ szubalgoritmusok szerkezetére vonatkozó 1—3. feltételek fennállása esetén az optimumbecslések $\{f_k\}$ sorozata f^*-hoz tart (m.m.). Emellett az $\{\bar{\mathbf{x}}_k\}$ sorozat bármely $\{\bar{\mathbf{x}}_{k_m}\}$ konvergens részsorozata a feladat egy $\bar{\mathbf{x}}$ globális megoldásához tart (m.m.), vagyis $\bar{\mathbf{x}}\in C$ és $f(\bar{\mathbf{x}})=f^*$ (m.m.).

Bizonyítás. Először is belátjuk, hogy minden $\varepsilon>0$ számhoz létezik olyan $k_0=k_0(\varepsilon)$ index és $p_0=p_0(\varepsilon)>0$ érték, hogy $k\geqq k_0$ esetén érvényes a

$$P(f_k>f^*+\varepsilon)\leqq\exp\left(-p_0\sum_{j=k_0}^{k}\alpha_j\right) \tag{3.7}$$

reláció; ebből azután könnyen adódik a tétel állítása.

Legyen $k_0^{(1)}$ olyan index, hogy $k\geqq k_0^{(1)}$ esetén már teljesül $\varepsilon_k<\varepsilon$: ekkor a $G_k(\varepsilon_k)$ $k\geqq k_0^{(1)}$ szubalgoritmusok bármelyike lehetővé teszi ε-optimális megoldás megtalálását $p_k(\varepsilon)$-nál nem kisebb valószínűséggel. A (3.5) reláció értelmében létezik olyan $k_0^{(2)}$ index és $0<p_0\leqq 1$ szám, amelyekkel $p_k(\varepsilon)\geqq p_0$, ha $k\geqq k_0^{(2)}$. Legyen $k_0=\max(k_0^{(1)}, k_0^{(2)})$, ekkor $k\geqq k_0$ esetén fennáll

$$P(f_k>f^*+\varepsilon)\leqq P\left(\bigcap_{j=k_0}^{k}\{f(\mathbf{x}_j)>f^*+\varepsilon\}\right)= \tag{3.8}$$
$$=\prod_{j=k_0}^{k}P(\{f(\mathbf{x}_j)>f^*+\varepsilon\}|\{f(\mathbf{x}_i)>f^*+\varepsilon\},\quad i=k_0,\ldots,j-1).$$

Itt viszont a $\mathbf{G}_k$ szubalgoritmusok megválasztása folytán érvényes a következő becslés:

$$P(\{f(\mathbf{x}_j)\leqq f^*+\varepsilon\}|\{f(\mathbf{x}_i)>f^*+\varepsilon\},\quad i=k_0,\ldots,j-1)= \tag{3.9}$$
$$=P[\{f(\mathbf{x}_j)\leqq f^*+\varepsilon\}|(\{f(\mathbf{x}_i)>f^*+\varepsilon\},\quad i=k_0,\ldots,j-1);\ \mathbf{G}_j]\cdot\alpha_j+$$
$$+P[\{f(\mathbf{x}_j)\leqq f^*+\varepsilon\}|(\{f(\mathbf{x}_i)>f^*+\varepsilon\},\quad i=k_0,\ldots,j-1);\ \mathbf{L}_j]\cdot(1-\alpha_j)\geqq$$
$$\geqq p_j(\varepsilon)\cdot\alpha_j,$$

ezért (3.8) és (3.9) alapján adódik a bizonyítandó egyenlőtlenség:

$$P(f_k>f^*+\varepsilon)\leqq\prod_{j=k_0}^{k}(1-p_j(\varepsilon)\cdot\alpha_j)\leqq\prod_{j=k_0}^{k}(1-p_0\alpha_j)\leqq \tag{3.10}$$
$$\leqq\exp\left(-p_0\sum_{j=k_0}^{k}\alpha_j\right).$$

A (3.10) relációból $\sum_{k=0}^{\infty} \alpha_k = \infty$ figyelembevételével azonnal következik

$$\lim_{k\to\infty} P(f_k > f^* + \varepsilon) = 0 \quad \text{(minden } \varepsilon > 0 \text{ esetén)},$$

vagyis $\{f_k\}$ sztochasztikusan konvergál az f^* értékhez. Az $\{f_k\}$ sorozat monotonítása miatt $P(\lim_{k\to\infty} f_k = f^*) = 1$ is teljesül, azaz $\lim_{k\to\infty} f_k = f^*$ (m.m.).

Tekintsük $\{\bar{\mathbf{x}}_k\}$-nak egy konvergens $\{\bar{\mathbf{x}}_{k_m}\}$ részsorozatát. Mivel C zárt, bármely ilyen részsorozat esetén $\bar{\mathbf{x}} = \lim_{m\to\infty} \bar{\mathbf{x}}_{k_m}$ eleme C-nek és $f(\mathbf{x})$ folytonossága miatt teljesül az $f(\bar{\mathbf{x}}) = \lim_{m\to\infty} f(\bar{\mathbf{x}}_{k_m}) = \lim_{k\to\infty} f_k = f^*$ (m.m.) reláció is.

Tekintsük most az alábbi, feltételekkel korlátozott, sztochasztikus optimalizálási feladatot:

$$\min_{\mathbf{x}\in C} f(\mathbf{x}) \quad f(\mathbf{x}) \approx F(\mathbf{x}, \xi), \quad C \approx C(\eta) \tag{3.11}$$

A (3.11) probléma és a megoldó eljárás szerkezetére vonatkozóan a következő feltevésekkel élünk (vö. a korábbi tételek feltételeivel):

Az iterációs lépések során mind a célfüggvény, mind pedig a megengedett megoldások halmaza előállítható determinisztikus és sztochasztikus komponens összegeként

$$F(\mathbf{x}_k, \xi) = f(\mathbf{x}_k) + \xi(\mathbf{x}_k) \tag{3.12}$$

$$C(\eta) = C(\mathbf{x}_k, \eta) = C + \eta(\mathbf{x}_k),$$

és bármely $\{\mathbf{x}_k\}$ generált pontsorozat esetén (ω szerint egyenletesen) teljesülnek az alábbi feltételek:

1. *a)* $f(\mathbf{x})$, $\mathbf{x}\in C(\eta)\subset E^n$ folytonos, konstanstól különböző függvény;
   b) $\xi(\mathbf{x}, \omega)$: $\Omega \to E^1$ (minden $\mathbf{x}\in E^n$ esetén) valószínűségi változó és fennáll $\left|\sum_{k=0}^{\infty} \xi(\mathbf{x}_k, \omega)\right| \leqq S$ (m.m.) ($0 < S < \infty$ — állandó).

2. *a)* C valamely nem-üres, korlátos, összefüggő, nyílt halmaz lezártja;

   b) $\eta(\mathbf{x}, \omega)$: $\Omega \to 2^{E^n}$ *P-mérhető* véletlen pont-halmaz leképezés, vagyis adott $\mathbf{x}\in E^n$ és $\omega\in\Omega$ esetén $\eta(\mathbf{x}, \omega)$ E^n valamely részhalmaza. Emellett fennáll — az $\|\eta(\mathbf{x}, \omega)\| = \operatorname{vrai\,sup}_{\mathbf{z}\in\eta(\mathbf{x},\omega)} \|\mathbf{z}\|$ definícióval — a $\sum_{k=0}^{\infty} \|\eta(\mathbf{x}_k, \omega)\| \leqq T$(m.m.) reláció ($0 < T < \infty$-állandó). Feltesszük még, hogy $C(\eta)$ maga is nem-üres, összefüggő, nyílt halmaz lezártja (korlátossága már a korábbi feltételekből következik).

3. Az $\{\mathbf{x}_k\}$ $\mathbf{x}_k\in C(\eta)$, $k = 0, 1, 2, \ldots$ sorozatot a (3.3) formula által leírthoz hasonló, sztochasztikusan kombinált módszerrel állítjuk elő. Itt feltesszük, hogy bármely[1] $\varepsilon > 0$ esetén a $\mathbf{G}_k = \mathbf{G}_k(\varepsilon)$ szubalgoritmus véges számú lépésben, legalább $0 < p_k(\varepsilon) \leqq 1$ valószínűséggel lehetővé teszi az ε-optimális halmaz k-adik közelítése, vagyis az

$$X_{\varepsilon,k} = \{\mathbf{x}_{\varepsilon,k}\in C + \eta(\mathbf{x}_k): f(\mathbf{x}_{\varepsilon,k}) \leqq f^* + \varepsilon\} \tag{3.13}$$

halmaz valamely pontjának kiválasztását. (Hangsúlyozzuk, hogy a véletlen zajhatások miatt egy ilyen pont ε-optimális voltát az eljárás nem szükségképpen érzékeli.) A korábbiakhoz hasonlóan feltesszük, hogy $p_k(\varepsilon)$ ε monoton növekvő függvénye $k=0, 1, 2, \ldots$ esetén, továbbá minden $\varepsilon>0$ és generált $\{\mathbf{x}_k\}$ sorozat esetén teljesül a (3.5) reláció. Az $\mathbf{L}_k$ (tetszőlegesen választható) szubalgoritmusok szerepe a korábbival megegyező.

Az iteratív hibrid eljárást a determinisztikus esethez hasonlóan konstruáljuk meg: $\mathbf{G}_k=\mathbf{G}_k(\varepsilon_k)$ $\varepsilon_k>0$, $k=0, 1, 2, \ldots$ és $\lim_{k\to\infty}\varepsilon_k=0$. Ebben az esetben az optimum és az optimális megoldás k-adik közelítését az

$$f_{k,\varepsilon_k}=\min_{\substack{0\leqq j\leqq k\\ |\xi(x_j)|\leqq\varepsilon_k}}\{F(\mathbf{x}_j,\xi)\}=F(\bar{\mathbf{x}}_{k,\varepsilon_k},\xi) \tag{3.14}$$

képlettel definiáljuk; itt előírjuk, hogy az $\bar{\mathbf{x}}_{k,\varepsilon_k}$ pontbeli függvényérték, vagyis az aktuális optimumbecslés maximális hibája ε_k legyen ($\bar{\mathbf{x}}_{k,\varepsilon_k}$ egyértelmű kiválasztását a korábbi megállapodással biztosíthatjuk). Érvényes a 3.1. tételnek a (3.11) sztochasztikus probléma esetére vonatkozó megfelelője:

3.2. TÉTEL. A (3.11) feladat és az alkalmazható $\mathbf{G}_k$ globális szubalgoritmusok struktúrájára vonatkozó 1—3. feltételek fennállása esetén az optimumbecslések $\{f_{k,\varepsilon_k}\}$ sorozata a (3.11)-be „beágyazott" determinisztikus feladat f^* optimumához konvergál (m.m.). Emellett az $\{\bar{\mathbf{x}}_{k,\varepsilon_k}\}$ sorozat korlátos (m.m.) és bármely konvergens $\{\bar{\mathbf{x}}_{k_m}\}$ részsorozata a megfelelő determinisztikus feladat valamely $\bar{\mathbf{x}}$ globális megoldásához tart, vagyis $\bar{\mathbf{x}}\in C$ és $f(\bar{\mathbf{x}})=f^*$ (m.m.).

Bizonyítás. Először a (3.7) egyenlőtlenséghez hasonló relációt igazolunk. Ehhez megjegyezzük, hogy a $\sum_{k=0}^{\infty}\xi(\mathbf{x}_k)$ sor (ω szerint egyenletes) konvergenciáját figyelembevéve minden $\varepsilon>0$ esetén létezik olyan $k_0^{(3)}=k_0^{(3)}(\varepsilon)$ index, hogy $k\geqq k_0^{(3)}$ esetén $|\xi(\mathbf{x}_k)|<\varepsilon$ (m.m.). Ezért $k\geqq k_0^{(3)}$ esetén (m.m.) igaz $F(\mathbf{x}_k,\xi)>f^*-\xi(\mathbf{x}_k)>f^*-\varepsilon$, emellett minden k-ra fennáll

$$f_{k,\varepsilon_k}>f^*-\varepsilon_k, \tag{3.15}$$

továbbá, ha $f(\mathbf{x}_k)\leqq f^*+\varepsilon$, akkor teljesül (m.m.)

$$F(\mathbf{x}_k,\xi)\leqq f^*+2\varepsilon. \tag{3.16}$$

Legyen ezért tetszőleges adott $\varepsilon>0$ mellett $k_0=k_0(\varepsilon)=\max\{k_0^{(1)}, k_0^{(2)}, k_0^{(3)}\}$, ahol $k\geqq k_0^{(1)}$ esetén $\varepsilon_k<\varepsilon$, $k\geqq k_0^{(2)}$ esetén pedig $p_k(\varepsilon)\geqq p_0=p_0(\varepsilon)>0$ (vö. a (3.7) reláció bizonyításával). Ekkor $k\geqq k_0$ esetén teljesül

$$P(f_{k,\varepsilon_k}>f^*+2\varepsilon)=P\Big(\min_{\substack{0\leqq j\leqq k\\ |\xi(x_j)|\leqq\varepsilon_k}}F(\mathbf{x}_j,\xi)>f^*+2\varepsilon\Big)\leqq \tag{3.17}$$

$$\leqq P\Big(\bigcap_{j=k_0}^{k}\{f(\mathbf{x}_j)>f^*+\varepsilon\}\Big)\leqq\exp\Big(-p_0\sum_{j=k_0}^{k}\alpha_j\Big),$$

ahol az utolsó egyenlőtlenség a (3.8)—(3.10) relációkból adódik.

A (3.15) és (3.17) egyenlőtlenségekből $\sum_{k=0}^{\infty} \alpha_k = \infty$ esetén nyilvánvaló, hogy $\{f_{k,\varepsilon_k}\}$ sztochasztikusan konvergál f^*-hoz. Az $\{f_{k,\varepsilon_k}\}$ sorozat definíciója értelmében az 1 valószínűségű konvergenciára vonatkozó állítás is adódik. Továbbá, mivel $\mathbf{x}_k \in C(\eta)$, $k=0, 1, 2, \ldots$ és $\sum_{k=0}^{\infty} \|\eta(\mathbf{x}_k)\| < \infty$, következik, hogy $\{\bar{\mathbf{x}}_{k,\varepsilon_k}\} \subset \{\mathbf{x}_k\}$ — korlátos pontsorozat. Ebből konvergens $\{\bar{\mathbf{x}}_{k_m}\}$ $m=1, 2, \ldots$ részsorozatot kiválasztva, a $\lim_{k\to\infty} \|\eta(\mathbf{x}_k)\| = 0$ relációból és a C halmaz zártságából következik, hogy $\bar{\mathbf{x}} = \lim_{m\to\infty} \bar{\mathbf{x}}_{k_m} \in C$. Végül $f(\mathbf{x})$ folytonossága és a $\lim_{k\to\infty} \xi(\mathbf{x}_k) = 0$ reláció alapján az $f(\bar{\mathbf{x}}) = \lim_{m\to\infty} f(\bar{\mathbf{x}}_{k_m}) = \lim_{k\to\infty} f_{k,\varepsilon_k} = f^*$ egyenlőség is következik.

4. Összefoglalás

Dolgozatunkban sztochasztikusan kombinált (hibrid) optimalizálási eljárások egy osztályának lokális és globális konvergenciatulajdonságait vizsgáltuk. A bizonyításokból látható, hogy az elméleti konvergenciát a leírt típusú $\{\mathbf{R}_k\}$ és $\{\mathbf{G}_k\}$ szubalgoritmusok az $\{\mathbf{L}_k\}$ lokális szubalgoritmusoktól függetlenül is biztosítják. Ezért konkrét hibrid módszerek kiválasztása a megoldandó feladat (várható) strukturális sajátosságai alapján történhet. Itt nem bocsátkozunk ennek a gyakorlati szempontból igen fontos kérdéskörnek a részleteibe, csupán megemlítjük, hogy az $\{\mathbf{L}_k\}$ szubalgoritmusok szerepét pl. a már idézett konjugált gradiens típusú módszerek valamelyike játszhatja. Emellett az $\{\mathbf{R}_k\}$ módszerek úgy választhatók meg, hogy teljesüljenek a (2.5) és (2.7) relációk megfelelői (pl. a korábban leírt módon $\mathbf{R}_k = \mathbf{R}$, $k=0, 1, 2, \ldots$ lehet az irányválasztási stratégia, megfelelő lokális iránymenti közelítő optimalizálási eljárással kiegészítve). A $\{\mathbf{G}_k\}$ módszerek pl. az $\mathbf{R}$ stratégiát valamely globális iránymenti minimalizálási eljárással (l. pl. [22, 34]) kiegészítve definiálhatók.

Megjegyezzük még, hogy a bizonyított tételek néhány feltételét valamelyest enyhíteni lehet. Pl. a 2.2. és 3.2. tételek bizonyításából látható, hogy a $\sum_{k=0}^{\infty} \|\eta(\mathbf{x}_k, \omega)\| < \infty$ feltétel helyett elégséges a $\lim_{k\to\infty} \|\eta(\mathbf{x}_k, \omega)\| = 0$ reláció megkövetelése. (A $\sum_{k=0}^{\infty} \xi(\mathbf{x}_k, \omega)$, $\sum_{k=0}^{\infty} \varepsilon(\mathbf{x}_k, \omega)$, $\sum_{k=0}^{\infty} \|\eta(\mathbf{x}_k, \omega)\|$ sorok konvergenciájára vonatkozóan pl. *Kolmogorov háromsor-tételéből* adódik feltétel, l. [21], 351—356.) Egyszerűen adódik továbbá, hogy a (2.7), ill. (3.5) képletekben szereplő $p>0$, ill. $p(\varepsilon)>0$ létezése helyett elegendő feltenni olyan $\{p_k\}$, ill. $\{p_k(\varepsilon)\}$ sorozatok létezését, amelyekre teljesül $\sum_{k=0}^{\infty} p_k \beta_k = \infty$, ill. $\sum_{k=0}^{\infty} p_k(\varepsilon)\alpha_k = \infty$. Ez közvetlenül látható a (2.7), ill. (3.9) relációk alapján.

5. Függelék

Az alábbiakban az (1.1) sztochasztikus programozási feladat globális megoldásának létezésére vonatkozó állítást igazolunk.

5.1. TÉTEL. Tegyük fel, hogy az (1.1) feladat megengedett megoldásainak halmaza nem-üres, a szereplő $K \subset E^n$ halmaz korlátos és zárt, az $f(\mathbf{x}, \mathbf{z})$ és $g_i(\mathbf{x}, \mathbf{z})$, $i=1, \ldots, m$ függvények az $E^n \times E^q$ szorzattérben $\mathbf{x}$ szerint egyenletesen folytonosak, továbbá az $\mathbf{y}$ valószínűségi vektorváltozó abszolút folytonos eloszlásfüggvénnyel rendelkezik. Ha az $\mathbf{y}$ — adott $\mathbf{x}$ mellett — megengedett realizációi halmazának (l. az (5.1)-ben szereplő $A(\mathbf{x})$ halmazt) mértéke az (5.3) reláció szerint $\mathbf{x}$ folytonos függvénye, akkor az (1.1) feladatnak létezik globálisan optimális megoldása.

Bizonyítás. Legyen $h(\mathbf{z})$ az $\mathbf{y}$ valószínűségi változó sűrűségfüggvénye. Vezessük be az

$$F(\mathbf{x}) = M_y[f(\mathbf{x}, \mathbf{y})] = \int_{E^q} f(\mathbf{x}, \mathbf{z}) h(\mathbf{z})\, d\mathbf{z}$$

$$A(\mathbf{x}) = \{\mathbf{z} \in E^q : g_i(\mathbf{x}, \mathbf{z}) \geqq 0,\ i = 1, \ldots, m\} \quad (\mathbf{x} \in K) \tag{5.1}$$

$$G(\mathbf{x}) = \int_{A(\mathbf{x})} h(\mathbf{z})\, d\mathbf{z}$$

jelöléseket. Igazoljuk, hogy $F(\mathbf{x})$ és $G(\mathbf{x})$ folytonos függvények: ekkor ugyanis a $K \cap \{\mathbf{x}: G(\mathbf{x}) \geqq p\}$ megengedett halmaz (nem-üres), korlátos és zárt, tehát az analízis ismert ([30]-ból idézett) tétele alapján $F(\mathbf{x})$ felveszi minimumát a megengedett halmaz valamelyik pontjában.

Az $f(\mathbf{x}, \mathbf{z})$ függvény $\mathbf{x}$ szerinti egyenletes folytonossága alapján nyilvánvaló, hogy bármely $\varepsilon_1 > 0$ esetén igaz

$$|F(\mathbf{x}_1) - F(\mathbf{x}_2)| \leqq \int_{E^q} |f(\mathbf{x}_1, \mathbf{z}) - f(\mathbf{x}_2, \mathbf{z})| h(\mathbf{z})\, d\mathbf{z} \leqq \tag{5.2}$$

$$\leqq \operatorname*{vrai\,sup}_{\mathbf{z} \in E^q} |f(\mathbf{x}_1, \mathbf{z}) - f(\mathbf{x}_2, \mathbf{z})| < \varepsilon_1,$$

ha $\|\mathbf{x}_1 - \mathbf{x}_2\| < \delta_1 = \delta_1(\varepsilon_1)$: F tehát $\mathbf{x}$-nek folytonos függvénye.

Másrészt, az $A(\mathbf{x})$ halmaz mértékének $\mathbf{x}$ szerinti folytonossága alapján bármely $\varepsilon_2 > 0$ esetén az $A(\mathbf{x}_1)$ és $A(\mathbf{x}_2)$ halmazok szimmetrikus differenciájának *Lebesgue-mértékére* teljesül

$$\mu[A(\mathbf{x}_1) \Delta A(\mathbf{x}_2)] < \varepsilon_2, \tag{5.3}$$

ha $\|\mathbf{x}_1 - \mathbf{x}_2\| < \delta_2 = \delta_2(\varepsilon_2)$. Ezért bármely $\varepsilon_3 > 0$ esetén igaz az

$$|G(\mathbf{x}_1) - G(\mathbf{x}_2)| = \left| \int_{A(\mathbf{x}_1)} h(\mathbf{z})\, d\mathbf{z} - \int_{A(\mathbf{x}_2)} h(\mathbf{z})\, d\mathbf{z} \right| \leqq \tag{5.4}$$

$$\leqq \int_{A(\mathbf{x}_1) \Delta A(\mathbf{x}_2)} h(\mathbf{z})\, d\mathbf{z} < \varepsilon_3$$

egyenlőtlenség is, ha $\mu[A(\mathbf{x}_1) \Delta A(\mathbf{x}_2)] < \varepsilon_2 = \varepsilon_2(\varepsilon_3)$: tehát G is folytonos függvénye $\mathbf{x}$-nek.

IRODALOM

[1] BLUM, J. R., "Multidimensional stochastic approximation methods", *Annals of Mathematical Statistics* **25** (1954) 737—744.

[2] DEVROYE, L. P., "Progressive global random search of continuous functions", *Mathematical Programming* **15** (1978) 330—342.

[3] FLETCHER, R. and REEVES, C. M., "Function minimization by conjugate gradients", *Computer Journal* **2** (1964) 149—154.

[4] GERENCSÉR, L., „Nemlineáris programozási feladatok megoldása szekvenciális módszerekkel", (Kandidátusi disszertáció, Budapest, 1975), *MTA SZTAKI Tanulmányok* **49** (1976).

[5] HESTENES, M. R. and STIEFEL, E., "Methods of conjugate gradients for solving linear systems", *Journal of Researches of the National Bureau of Standards* **49** (1952) 409—436.

[6] HIMMELBLAU, D. M., "A uniform evaluation of unconstrained optimization techniques", in: *Lootsma, F. A.* (Editor): *Numerical Methods for Nonlinear Optimization* (London, Academic Press, 1972). 69—97.

[7] KUSHNER, H. J. and KELMANSON, M. L., "Stochastic approximation algorithms of the multiplier type for the sequential Monte Carlo optimization of stochastic systems", *SIAM Journal on Control and Optimization* **14** (1976) 827—842.

[8] MAYER, J., „A STABIL sztochasztikus programozási modellről", *Alkalmazott Matematikai Lapok* **2** (1976) 171—187.

[9] PINTÉR, J., „A Sajó-térség vízgazdálkodási modellje", *Vízügyi Közlemények* (1977) 418—427.

[10] PINTÉR, J., „Véletlen kereső eljárások konvergenciájának és numerikus hatékonyságának vizsgálata", *Alkalmazott Matematikai Lapok* **4** (1978) 197—228.

[11] PINTÉR, J., "On a stochastic model of reservoir system sizing", (Proceedings of the 9th IFIP Conference on Optimization Techniques, 4—8. September, 1979, Warszawa) *Lecture Notes in Control and Information Sciences* **23** (Berlin—Heidelberg, Springer, 1980) 546—558.

[12] PINTÉR, J., "Regionális vízminőségvédelmi döntési problémák sztochasztikus modelljei", *Hidrológiai Közlöny* (1980) 364—373.

[13] POLJAK, B. T., "Nonlinear programming methods in the presence of noise", *Mathematical Programming* **14** (1978) 87—97.

[14] POWELL, M. J. D., "An efficient method of finding the minimum of a function of several variables without calculating derivatives", *Computer Journal* **7** (1964) 155—162.

[15] PRÉKOPA, A., Sztochasztikus rendszerek optimalizálási problémáiról (Akadémiai doktori disszertáció, Budapest, 1970).

[16] PRÉKOPA, A., GANCZER, S., DEÁK, I. és PATYI, K., „A STABIL sztochasztikus programozási modell és annak kísérleti alkalmazása a magyar villamosenergia-iparra", *Alkalmazott Matematikai Lapok* **1** (1975) 3—22.

[17] PRÉKOPA, A., RAPCSÁK, T. és ZSUFFA, I., „Egy új módszer sorbakapcsolt tározórendszer tervezésére sztochasztikus programozás felhasználásával", *Alkalmazott Matematikai Lapok* **2** (1976) 189—201.

[18] PRÉKOPA, A. és SZÁNTAI, T., „Árvízi tározók méretezése sztochasztikus programozással", *Alkalmazott Matematikai Lapok* **2** (1976) 203—217.

[19] PRÉKOPA, A. és SZÁNTAI, T., „Készletszintek optimális szabályozása és annak alkalmazása a Balaton vízszintszabályozására", *Hidrológiai Közlöny* (1980) 392—399.

[20] RAPCSÁK, T., „A SUMT módszer alkalmazása nem konvex programozási feladatok esetén", *Alkalmazott Matematikai Lapok* **2** (1976) 427—437.

[21] RÉNYI, A., *Valószínűségszámítás* (Budapest, Tankönyvkiadó, 1968).

[22] SHUBERT, B. O., "A sequential method seeking the global maximum of a function", *SIAM Journal on Numerical Analysis* **9** (1972) 379—388.

[23] SLOBODA, F., "Nonlinear iterative methods and parallel computation", *Aplikace Matematiky* **21** (1976) 252—262.

[24] SZÁNTAI, T., „A Prékopa-féle STABIL sztochasztikus programozási modell numerikus megoldásáról", *Alkalmazott Matematikai Lapok* **2** (1976) 93—101.

[25] Айзерман, М. А., Браверман, Э. М. и Розоноэр, Л. И., *Метод потенциальных функций в теории обучения машин* (Москва, Наука, 1970).

[26] Беленький, В. З., Волконский, В. А., Иванков, С. А., Поманский, А. Б. и Шапиро, А. Д., *Итеративные методы в теории игр и программирования* (Москва, Наука, 1974).

[27] Богомолов, Н. А. и Карманов, В. Г., «О сходимости метода случайного поиска для отыскания стационарных точек общей задачи нелинейного программирования», *Вестник Московского Университета, Серия 15. Вычислительная математика и кибернетика* (1977) №. 1., 20—26.
[28] Денисов, Д. В., «О методе случайного поиска в задачах условной минимизации», *Журнал вычислительной математики и математической физики* **18** (1978) 1103—1111.
[29] Ермольев, Ю. М., *Методы стохастического программирования* (Москва, Наука, 1976).
[30] Колмогоров, А. Н. и Фомин, С. В., *Элементы теории функций и функционального анализа* (Москва, Наука, 1968).
[31] Невельсон, М. Б. и Хасьминский, Р. З., *Стохастическая аппроксимация и рекуррентное оценивание* (Москва, Наука, 1972).
[32]. Поляк, Б. Т., «Сходимость и скорость сходимости итеративных стохастических алгоритмов, 1. Общий случай», *Автоматика и телемеханика* (1976) №. 12., 83—94.
[33] Растригин, Л. А., *Системы экстремального управления* (Москва, Наука, 1974).
[34] Стронгин, Р. Г., *Численные методы в многоэкстремальных задачах* (Москва, Наука, 1978).

(Beérkezett: 1981. február 2.)

PINTÉR JÁNOS
ELTE SZÁMÍTÓKÖZPONT
1093 BUDAPEST, DIMITROV TÉR 8.

HYBRID OPTIMIZATION PROCEDURES FOR THE SOLUTION OF NON-SMOOTH STOCHASTIC PROBLEMS

J. PINTÉR

A general class of stochastically combined optimization procedures is investigated, theorems on their local and global convergence properties are proved. These procedures are applicable e.g. for solving non-smooth (continuous) mathematical programming problems, in the presence of noise.

Alkalmazott Matematikai Lapok 7 (1981) 99—105

LINEÁRIS PROGRAMOZÁSI MODELL EGY TEREPRENDEZÉSI FELADAT MEGOLDÁSÁRA

RAPCSÁK TAMÁS

Budapest

A cikkben szereplő modell a tereprendezési feladatok egy típusának optimális megoldását nagyméretű lineáris programozási feladatra vezeti vissza. A modell előnye, hogy az elkészített mátrix generáló program segítségével tetszőleges alakú és nagyságú terület is könnyen kezelhető. A módszert egy gyakorlati probléma megoldásakor próbáltuk ki, ahol a lineáris programozási feladat mátrixának 1359 sora volt.

1. Bevezetés

Egy új létesítmény megvalósítása során az első lépés mindig a tereprendezési feladat megoldása. Ez általában igen időigényes, sok fáradságot igénylő feladat, amely nagy volumenű földmennyiség megmozgatását teszi szükségessé.

Az ilyen típusú feladatokat több csoportba lehet osztani. Az egyikbe azok tartoznak, amelyek öntözővíz hálózatok létesítésekor is fellépnek. Mi a továbbiakban csak ezekkel foglalkozunk. Itt a probléma a következő:

Adott n számú téglalap alakú parcella, amelyek további kisebb téglalapokra vannak osztva. Minden kis téglalap középpontjában ismert a terep magassága. A feladat a víz sebességének, illetve a lejtések alsó és felső határainak ismeretében egy olyan műterep kialakítása, azaz olyan új terepmagasságok meghatározása, amelyek esetén a szükséges földmunka mennyisége minimális, a föld feltöltések és a nyesések mennyisége körülbelül egyenlő. (A lejtés és sebességhatárok úgy vannak megadva, hogy az öntözővíz gravitációs úton a terep minden pontjához elérjen.) Ezenkívül figyelembe kell venni a különböző parcellák közötti kapcsolatokat is.

Az eddigi gyakorlat szerint a feladat megoldása egy kiegyenlítő sík megadásával vagy az úgynevezett négypontos módszerrel történt. Ez utóbbi esetben minden kis téglalap esetén egy megfelelő kiegyenlítő síkot kell meghatározni. Ezek a módszerek azonban nem biztos, hogy optimális megoldást adnak és a parcellák közötti kapcsolatok figyelembe vétele is körülményes.

A tereprendezési munkák elvégzésénél gyakran egy szállítási feladat is fellép, amelynek a megoldása egy következő lépésben történik ([4]).

Az itt bemutatott modell a feladatot nagyméretű lineáris programozási feladatra vezeti vissza, amelybe szükség esetén a szállítási feladat is beépíthető. A modellt az EL HADJIRA-i öntözőterület tereprendezési munkáinak számításakor próbáltuk ki, a számításokat az Algériai Vízügyi Minisztérium Számítóközpontjának UNIVAC 1106 gépén végeztük.[1]

[1] A munka a SETHYAL, az Algériai Vízügyi Minisztérium Tervező Intézetének a támogatásával folyt.

A lineáris programozási feladat megoldásához a UNIVAC gépekre kifejlesztett *Marie-Claire* nevű programcsomagot használtuk fel. A munka folyamán a legnagyobb nehézséget a mátrix generáló program megírása jelentette. Ez FORTRAN nyelven íródott.

A megoldott lineáris programozási feladat mátrixának 1359 sora volt. A megoldást kb. 10 perc alatt, 500—700 szimplex lépésben kaptuk meg. (Az eredeti feladatot több alkalommal módosítani kellett, mert a megengedett tartomány üres volt.)

Köszönetet mondok a SETHYAL-ban dolgozó magyar csoportnak és vezetőjének GARAMI F.-nek a probléma felvetéséért, értékes tanácsaikért és a konkrét feladat megoldásához nyújtott segítségükért.

2. A probléma és a modell

Tekintsünk egy tetszőleges alakú öntözendő területet. A mérnöki gyakorlatban elfogadott, hogy ezt a tartalmazó téglalapok méreteivel és a geodéziai felvételi háló rácspontjaiban mért magasságokkal kell leírni. (A tereprendezési munka a rácspontokban elvégzendő nyesésekkel és feltöltésekkel van megadva.) Így az általánosság megszorítása nélkül feltehetjük, hogy adott n számú téglalap alakú parcella, amelyek további kisebb téglalapokra vannak osztva. (Egy mezőgazdasági terület sok esetben már eleve parcellákra van osztva.) Minden kis téglalap középpontjában ismert a terep magassága. A feladat abban áll, hogy a középpontok új magasságait úgy határozzuk meg, hogy a következő feltételek teljesüljenek:

a) a terep lejtése minden parcellában, vízszintes és függőleges irányban a megadott korlátok között legyen,
b) a föld feltöltések és a nyesések összmennyisége legyen egyenlő,
c) nyesésnél a középpontok régi és új magasságainak a különbsége legyen felülről korlátos (hasonló kritérium megadható feltöltésre is),
d) a parcellák közötti kapcsolatok legyenek figyelembe véve,
e) az összes földmunka mennyisége legyen minimális.

A feladatban az *a)* feltétel azt biztosítja, hogy az öntözővíz gravitációs úton a terep minden pontjához elérjen. A *b)* kizárja azt, hogy a földet nagyobb távolságból vagy nagyobb távolságra kelljen szállítani. A *c)* feltételre azért van szükség, hogy a legfelső, termékeny földréteg megmaradjon, a *d)*-re pedig azért, hogy a terep sajátosságait figyelembe lehessen venni. Pl. ha két parcella között út van, vagy ha két vagy több parcella ugyanazt a szabálytalan területet fedi le, akkor a tereprendezés után a magassági szintek nem lehetnek lényegesen különbözőek.

Az itt szereplő modellben ez utóbbi feltételeket a terep ismeretében kiválasztott pontpárok magasság eltéréseire kiszabott alsó és felső korlátokkal adjuk meg.

Vezessük be a következő jelöléseket:

n a parcellák száma,

m a *d)* típusú feltételek száma,

$n_1(k)$ $k=1, \ldots, n$ a sorok száma a k-adik parcellában,

$n_2(k)$ $k=1, \ldots, n$ az oszlopok száma a k-adik parcellában,

$A(i, j, k)$ $i=1, \ldots, n_1(k), j=1, \ldots, n_2(k), k=1, \ldots, n$ a k-adik parcellában, az i-edik sorban és a j-edik oszlopban levő középpont magassága,

$B(j, k)$ $j=1, \ldots, n_2(k), k=1, \ldots, n$ a k-adik parcellában, a j-edik oszlopban a kis téglalapok vízszintes oldalainak hosszúsága,

$C(i, k)$ $i=1, \ldots, n_1(k), k=1, \ldots, n$ a k-adik parcellában, az i-edik sorban a kis téglalapok függőleges oldalainak hosszúsága,

$X(i, j, k)$ $i=1, \ldots, n_1(k), j=1, \ldots, n_2(k), k=1, \ldots, n$ a k-adik parcellának az i-edik sorában és a j-edik oszlopában a középpont magasságának megváltozása,

$EB1(i, j, k)$ $EB2(i, j, k)$, $i=1, \ldots, n_1(k)$, $j=1, \ldots, n_2(k)-1$, $k=1, \ldots, n$ a k-adik parcella i-edik sorában az (i, j) és $(i, j+1)$ középpontok között a műterep lejtésének alsó és felső határa,

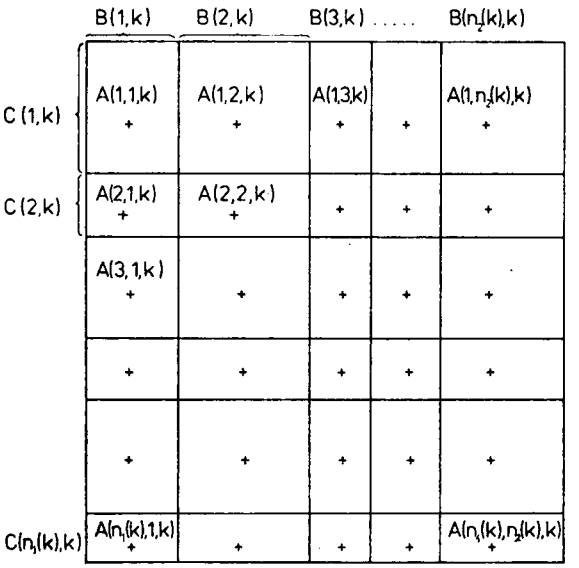
$EC1(i, j, k)$ $EC2(i, j, k)$, $i=1, \ldots, n_1(k)-1$, $j=1, \ldots, n_2(k)$, $k=1, \ldots, n$ a k-adik parcella j-edik oszlopában az (i, j) és $(i+1, j)$ középpontok között a műterep lejtésének alsó és felső határa,

$BA(i, j)$ $i=1, 2$, $j=1, \ldots, m$ a j-edik *d)* típusú feltételben az alsó és a felső határ,

RB a feltöltések összege,

DB a nyesések összege.

Az első ábrán példaképpen egy parcellát mutatunk be.



1. ábra

Az optimalizálási feladatot a következőképpen fogalmazzuk meg:

$$\min F(X)$$

$$EB1(i,j,k) \leqq \frac{2[A(i,j+1,k)+X(i,j+1,k)-A(i,j,k)-X(i,j,k)]}{B(j+1,k)+B(j,k)} \leqq EB2(i,j,k),$$

$$i = 1, \ldots, n_1(k), \quad j = 1, \ldots, n_2(k)-1, \quad k = 1, \ldots, n$$

$$EC1(i,j,k) \leqq \frac{2[A(i+1,j,k)+X(i+1,j,k)-A(i,j,k)-X(i,j,k)]}{C(i+1,k)+C(i,k)} \leqq EC2(i,j,k),$$

$$i = 1, \ldots, n_1(k)-1, \quad j = 1, \ldots, n_2(k), \quad k = 1, \ldots, n$$

$$-p_1 \cdot F(X) \leqq RB-DB \leqq p_2 \cdot F(X) \tag{2.1}$$

$$-X(i,j,k) \leqq p_3, \quad \text{ha} \quad X(i,j,k) < 0, \quad i = 1, \ldots, n_1(k), \quad j = 1, \ldots, n_2(k)$$

$$k = 1, \ldots, n$$

$$BA(1,l) \leqq A(i_1(l), j_1(l), k_1(l))+X(i_1(l), j_1(l), k_1(l))-A(i_2(l), j_2(l), k_2(l))-$$

$$-X(i_2(l), j_2(l), k_2(l)) \leqq BA(2,l), \quad l = 1, \ldots, m,$$

ahol

$$F(X) = \sum_{k=1}^{n} \sum_{i=1}^{n_1(k)} \sum_{j=1}^{n_2(k)} C(i,k) \cdot B(j,k) \cdot |X(i,j,k)|,$$

p_1, p_2, p_3 pedig a konkrét feladat ismeretében (gyakorlati szakemberek által) megfelelően választott paraméterek.

Látható, hogy a (2.1) probléma még nem lineáris programozási feladat, mert a célfüggvényben a változók abszolút értéke szerepel és az $RB-DB$ nincs kifejezve.

Megjegyezzük, hogy a modellben az RB és DB mennyiségek külön-külön nem kezelhetők, mert az optimalizálás minden lépésében változhat a bennük szereplő összeadandók száma.

Bevezetve az

$$X(i,j,k) = X_1(i,j,k)-X_2(i,j,k), \quad i = 1, \ldots, n_1(k), \quad j = 1, \ldots, n_2(k), \tag{2.2}$$

$$k = 1, \ldots, n$$

transzformációt, ahol $X_1(i,j,k)$ a pozitív részt, $X_2(i,j,k)$ pedig a negatív részt jelenti, a következő lineáris programozási feladatot kapjuk:

$$\min \sum_{k=1}^{n} \sum_{i=1}^{n_1(k)} \sum_{j=1}^{n_2(k)} C(i,k) \cdot B(j,k) \cdot X_1(i,j,k)+C(i,k) \cdot B(j,k) \cdot X_2(i,j,k)$$

$$\sum_{k=1}^{n} \sum_{i=1}^{n_1(k)} \sum_{j=1}^{n_2(k)} (p_1+1)C(i,k)B(j,k)X_1(i,j,k)+(p_1-1)C(i,k)B(j,k) \cdot X_2(i,j,k) \geqq 0,$$

$$\sum_{k=1}^{n}\sum_{i=1}^{n_1(k)}\sum_{j=1}^{n_2(k)}(p_2-1)\cdot C(i,k)B(j,k)X_1(i,j,k)+(p_2+1)C(i,k)B(j,k)\cdot$$
$$\cdot X_2(i,j,k)\geqq 0,$$

$$-X_1(i,j,k)+X_2(i,j,k)+X_1(i,j+1,k)-X_2(i,j+1,k)\leqq$$
$$\leqq EB2(i,j,k)\cdot\left(\frac{B(j+1,k)+B(j,k)}{2}\right)-\big(A(i,j+1,k)-A(i,j,k)\big)$$
$$i=1,\ldots,n_1(k),\quad j=1,\ldots,n_2(k)-1,\quad k=1,\ldots,n,$$

$$-X_1(i,j,k)+X_2(i,j,k)+X_1(i,j+1,k)-X_2(i,j+1,k)\geqq$$
$$\geqq EB1(i,j,k)\left(\frac{B(j+1,k)+B(j,k)}{2}\right)-\big(A(i,j+1,k)-A(i,j,k)\big)$$
$$i=1,\ldots,n_1(k),\quad j=1,\ldots,n_2(k)-1,\quad k=1,\ldots,n,$$

$$-X_1(i,j,k)+X_2(i,j,k)+X_1(i+1,j,k)-X_2(i+1,j,k)\leqq$$
$$\leqq EC2(i,j,k)\cdot\left(\frac{C(i+1,k)+C(i,k)}{2}\right)-\big(A(i+1,j,k)-A(i,j,k)\big) \tag{2.3}$$
$$i=1,\ldots,n_1(k)-1,\quad j=1,\ldots,n_2(k),\quad k=1,\ldots,n,$$

$$-X_1(i,j,k)+X_2(i,j,k)+X_1(i+1,j,k)-X_2(i+1,j,k)\geqq$$
$$\geqq EC1(i,j,k)\cdot\left(\frac{C(i+1,k)+C(i,k)}{2}\right)-\big(A(i+1,j,k)-A(i,j,k)\big)$$
$$i=1,\ldots,n_1(k)-1,\quad j=1,\ldots,n_2(k),\quad k=1,\ldots,n,$$

$$X_2(i,j,k)\leqq p_3,\quad i=1,\ldots,n_1(k),\quad j=1,\ldots,n_2(k),\quad k=1,\ldots,n,$$
$$X_1\big(i_1(l),j_1(l),k_1(l)\big)-X_2\big(i_1(l),j_1(l),k_1(l)\big)-X_1\big(i_2(l),j_2(l),k_2(l)\big)+$$
$$+X_2\big(i_2(l),j_2(l),k_2(l)\big)\geqq BA(2,l)-\big(A\big(i_1(l),j_1(l),k_1(l)\big)-A\big(i_2(l),j_2(l),k_2(l)\big)\big)$$
$$l=1,\ldots,m,$$

$$X_1\big(i_1(l),j_1(l),k_1(l)\big)-X_2\big(i_1(l),j_1(l),k_1(l)\big)-X_1\big(i_2(l),j_2(l),k_2(l)\big)+$$
$$+X_2\big(i_2(l),j_2(l),k_2(l)\big)\geqq BA(1,l)-\big(A\big(i_1(l),j_1(l),k_1(l)\big)-A\big(i_2(l),j_2(l),k_2(l)\big)\big)$$
$$l=1,\ldots,m,$$

$$\left.\begin{aligned}X_1(i,j,k)&\geqq 0\\ X_2(i,j,k)&\geqq 0\end{aligned}\right\}\quad i=1,\ldots,n_1(k),\quad j=1,\ldots,n_2(k),\quad k=1,\ldots,n.$$

Ez a modell egy nagyméretű lineáris programozási feladatot eredményez. Számítsuk ki hány sora és hány oszlopa lesz a feladat mátrixának. Ha az egyes par-

cellákban a műterep lejtéseire vonatkozó feltételek számát $s(k)$, $k=1, \ldots, n$ jelöli, akkor könnyű belátni, hogy

$$s(k) = (n_1(k)-1)(n_2(k)-1)\cdot 4+(n_1(k)-1)\cdot 2+(n_2(k)-1)\cdot 2, \quad k = 1, \ldots, n.$$

Mivel a felső korlát típusú feltételek száma megegyezik az összes középpont számával $\left(\sum_{k=1}^{n} n_1(k)\cdot n_2(k)\right)$, így az összes feltételek száma $\sum_{k=1}^{n} s(k)+\sum_{k=1}^{n} n_1(k)\cdot n_2(k)+2m+2$, a változók száma $\sum_{k=1}^{n} s(k)+\sum_{k=1}^{n} n_1(k)\cdot n_2(k)+2m+2+2\sum_{k=1}^{n} n_1(k)\cdot n_2(k)$.

3. A modell számítástechnikai megoldása és alkalmazása

Az előzőekben láttuk, hogy a modell egy nagyméretű lineáris programozási feladatot eredményez. A megoldás során a fő nehézséget a mátrix elemeinek megadott sorrendben történő generálása jelenti. Azonban a mátrix nemnulla együtthatóinak értéke minden feladat esetén ugyanaz (-1 vagy 1) kivéve azokat, amelyekben a p_1, p_2 paraméterek is szerepelnek. Így a p_1, p_2 ismeretében csak a nemnulla együtthatók pozícióit kell meghatározni.

A további könnyebbséget az jelenti, hogy a terület téglalap alakú parcellákból áll, illetve hogy ezek kisebb téglalapokra bonthatók. Ez lehetővé teszi, hogy a mátrix gépi realizálásánál a véges elem módszer ötletét alkalmazzuk, azaz ténylegesen csak egy (változtatható oldalhosszúságú) téglalap alakú parcella esetén határozzuk meg a -1, 1 együtthatók pozícióit. A többire ugyanezt az eljárást alkalmazva kapjuk meg a kívánt eredményt. Így tetszőleges területet véve, ugyanazzal a programmal, viszonylag kevés adat felhasználásával tudjuk a mátrix elemeit generálni.

A modellt az Algériai Vízügyi Minisztérium Számítóközpontjának UNIVAC 1106 gépén próbáltuk ki az EL HADJIRA-i öntözőterület tereprendezési munkáinak a számításakor. A lineáris programozási feladat megoldására az UNIVAC gépekre kifejlesztett *Marie-Claire* nevű programot használtuk fel. A mátrix generáló program FORTRAN nyelven íródott és a nemnulla együtthatókat oszlop folytonosan állítja elő.

A terület 9 parcellából állt, a középpontok száma 295 volt, a különböző parcellák közötti kapcsolatot 49 feltétel biztosította, így a lineáris programozási feladat mátrixának 1359 sora és 1949 oszlopa volt. A megoldást körülbelül 10 perc alatt, 500—700 szimplex lépésben kaptuk meg. (Az eredeti feladatot több alkalommal módosítani kellett, mert a megengedett tartomány üres volt.)

Megjegyezzük, hogy a megadható terület nagysága csak attól függ, hogy milyen méretű lineáris programozási feladatot tudunk megoldani.

4. A modell egy kiterjesztése

A tereprendezési munkák elvégzésénél gyakran egy szállítási probléma is fellép. Ennek a feladatnak számítógépen történő előállítása és megoldása a tereprendezés után következik ([4]). Itt röviden ismertetjük az előállítás elvét.

A geodéziai felvételi háló alapján készítsük el a szállítási hálót úgy, hogy az alábbi két szempont érvényesüljön:

a) a földszállítások gyakorlati végrehajtásával legyen összhangban,
b) a szállítási feladat mérete ne legyen túl nagy.

Ez azt jelenti, hogy a szállítási háló egy rácstéglalapja több kis téglalap egyesítéséből keletkezik. A rácstéglalapokon belül az összes nyeséseket, illetve feltöltéseket egy-egy súlypontba tömörítve kell figyelembe venni, ezért össze kell vonni az azonos jellegű tereprendezési földmunkákat, illetve ki kell számolni a súlypontok koordinátáit. Ezután a szállítási feladat költségmátrixát kell meghatározni.

A tapasztalatok szerint a módszer jól működik ([4]).

Mivel a bemutatásra került tereprendezési modell is lineáris programozási feladatot ad, ezért a két feladat összeépíthető. Így már a tereprendezési munkák tervezésekor figyelembe vehetők a szállítás szempontjai.

IRODALOM

[1] Bozóky-Szeszich, K., Kovács, K. és Illés, I., *Vízellátás és csatornázás* (Tervezési segédlet). Szerkesztette: *Dr. Öllős Géza* (Tankönyvkiadó, Bp., 1967).
[2] Carlier, M., *Hydraulique générale et appliqué* (Eyrolles, Paris, 1972).
[3] Fekete, I. és Dobos, A., *Az öntözés mezőgazdasági és műszaki tervezése* (Akadémiai Kiadó, Bp., 1972).
[4] Ijjas, I. és Földeáki, P., „Vízszintes síkú tereprendezés, új rizstelepek és rizstelep rekonstrukciók tervezése", *BME Vízgazdálkodási és Vízépítési Intézete*, 1980.

(Beérkezett: 1980. október 3.)

RAPCSÁK TAMÁS
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE
1250 BUDAPEST, URI U. 49.

A LINEAR PROGRAMMING MODEL FOR ORDERING OF THE TERRAIN

T. Rapcsák

In this paper a model is given for the optimal levelling of any kind of surface to be irrigated. The optimization problem is solved by a large scale linear programming model using the developed matrix generating program. In a practical application of the model the matrix had 1359 lines.

Alkalmazott Matematikai Lapok 7 (1981) 107—137

A FORGALOMSZÉTOSZTÁSI FELADAT OPTIMALIZÁCIÓS PROBLÉMÁI

BAKÓ ANDRÁS és KÁDAS SÁNDOR

Győr és Budapest

A közúti és városi úthálózatok számítógéppel segített fejlesztésének egyik legkritikusabb részfeladata a forgalom előrebecslési és szétosztási eljárás. A feladat megoldására sok eljárás ismert.

Dolgozatunkban összefoglaljuk a két fő módszertípust: a növekedési tényezős és a gravitációs típusú algoritmusokat. Bemutatjuk az ún. kalibrálási feladat megoldására kidolgozott eljárásunkat is. A gravitációs feladattal kapcsolatban megmutatjuk, hogy az *I*-divergencia minimalizálás és az entropia maximalizálás kritériuma ugyanarra a feladatra vezet.

Végül beszámolunk a számítógépes tapasztalatokról is.

1. Bevezetés

A forgalomszétosztási feladatban a jelenlegi forgalmi és egyéb adatokból a $\mathbf{H}=(h_{ij})$ ún. honnan-hová mátrix meghatározása a cél, melynek h_{ij} eleme az i-edik körzetből a j-edik körzetbe irányuló jelenlegi vagy jövőbeni forgalmat (adott időegységre, általában egy napra vetítve) adja meg.

A feladat megfogalmazása attól függ, hogy milyen adatok állnak rendelkezésre. Általában ismert a körzetekből kiinduló és a körzetekbe befutó forgalom. Ezt a jelenlegi forgalomnál forgalomszámlálásból, jövőbeni forgalomnál az előrebecslés első lépcsőjéből, az ún. forgalomkeltési eljárásból kapjuk meg. Ebből kell meghatározni azt a forgalmi mátrixot, amelyre az i-edik sor összege az i-edik körzetből kiáramló, a j-edik oszlop összege pedig a j-edik körzetbe befutó forgalmat adja meg.

Rendszerint rendelkezésünkre áll egy $\mathbf{G}=(g_{ij})$ ún. „minta" mátrix, a mért forgalmi mátrix, amelynek g_{ij} eleme az i-edik körzetből a j-edik körzetbe menő forgalmat jelenti. Jelölje a $\mathbf{G}$ mátrix sorösszegeit $o_1, o_2, \ldots, o_p$, melyek az $1, 2, \ldots, p$ körzetek kiáramló forgalmát jelentik. Az $1, 2, \ldots, q$ körzetekbe befutó forgalmakat, tehát a $\mathbf{G}$ mátrix oszlopösszegeit jelöljük $d_1, d_2, \ldots, d_q$-val. Így az o_i, d_j értékekre teljesül:

$$o_i = \sum_{j=1}^{q} g_{ij}, \quad i = 1, 2, \ldots, p, \tag{1.1}$$

és

$$d_j = \sum_{i=1}^{p} g_{ij}, \quad j = 1, 2, \ldots, q. \tag{1.2}$$

Számítással, különböző területfelhasználási adatok segítségével meghatározzuk körzetenként az előrebecslendő $\mathbf{H}$ mátrixra az $\bar{o}_i$ kimenő és $\bar{d}_j$ befutó forgalmakat. Az

$\bar{o}_i$ és $\bar{d}_j$ értékekre természetesen teljesülni kell a

$$\sum_{i=1}^{p} \bar{o}_i = \sum_{j=1}^{q} \bar{d}_j$$

feltételeknek. Az $\bar{o}_i$, $\bar{d}_j$ értékek jelenthetik a jelenlegi tényforgalmat vagy a jövőbeni forgalom nagyságát. A feladat ezek után meghatározni azt a $\mathbf{H}=(h_{ij})$ forgalmi mátrixot, amelyre

$$\sum_{j=1}^{q} h_{ij} = \bar{o}_i, \quad i = 1, 2, \ldots, p, \tag{1.3}$$

$$\sum_{i=1}^{p} h_{ij} = \bar{d}_j, \quad j = 1, 2, \ldots, q. \tag{1.4}$$

Bizonyos esetekben rendelkezésre áll a $\mathbf{C}=(c_{ij})$ költségmátrix, amelynek c_{ij} eleme az i-edik körzetből a j-edik körzetbe való utazás költségét jelenti. Mint a későbbiekben látni fogjuk, az előre megadott adatoktól függően a $\mathbf{H}$ mátrixra a fenti feltételeken kívül továbbiak is meg lehetnek adva.

Hogy a forgalomszétosztás szerepét jobban lássuk, ki kell térnünk röviden a forgalomelőrebecslési folyamat felépítésének, az egyes lépések szerepének a bemutatására is. A forgalomelőrebecslési folyamat egyes lépései személyforgalom esetén a következők:

a) *Forgalomkeltés*

Az egyes körzetek kiinduló és beérkező forgalmának előrebecslése a vizsgált jövőbeli időpontra: az $\bar{o}_i$, $\bar{d}_j$ értékek meghatározása. Itt célszerű közbülső lépcsőként a fajlagos (egy főre jutó) utazásszám becsléséből kiindulni. A fajlagos utazásszám becslése pedig leggyakrabban többváltozós regressziószámítás — legtöbbször lineáris regressziószámítás — segítségével történik. A változókat általában a körzetek szerkezeti mutatói — ún. városszerkezeti adatok —, pl. munkahelyek száma, kereskedelmi egységek alapterülete, átlagos háztartásméret, motorizációs szint, stb. adják.

b) *Forgalomszétosztás*

Erről a lépésről már szóltunk, s ennek változatait vizsgálja majd a cikk további része, így további részletezése itt nem szükséges.

c) *Közlekedési mód szerinti megosztás*

A forgalomszétosztás eredményeként kapott $\mathbf{H}$ mátrix h_{ij} elemei a körzetek közötti összes utazásszámot jelentik. A közlekedési hálózat terhelésének vizsgálata és egyéb szempontok miatt ezeket az utazásszámokat fel kell még osztani közlekedési módok (tömegközlekedés, egyéni közlekedés — személygépkocsis utak —, s esetleg

kerékpár- és gyalogutak) utazás számaira. Ennek különböző heurisztikus módszerei vannak, amelyek a múltbeli megoszlási adatok — megoszlási görbék —, s a befolyásoló tényezők kapcsolatának számszerűsítésére, valamint a jövőben várható ezirányú fejlődésre (pl. a motorizáció, a parkolási lehetőségek, a gépkocsihasználati szokások alakulása) támaszkodnak.

Sok esetben az előrebecslés első lépcsőjében a közlekedési mód szerinti megosztást a forgalomkeltéssel együtt, egyszerűsítve hajtják végre.

d) *A forgalom hálózati elosztása*

Ebben a lépésben történik meg az utazási mátrixok — külön a tömegközlekedésre, s az egyéni közlekedésre — hozzárendelése a közlekedési hálózathoz, a hálózat „terhelése". A modellel kapcsolatban azt a hipotézist szokás feltenni, hogy az utazók az utazási költségük minimalizálására törekszenek, azaz a leggazdaságosabb útvonalat veszik igénybe az utazásukra. Városi közlekedésnél az útvonal hosszának mérésénél az utazási idő a döntő tényező, ez pedig nyilván erősen függ a forgalom nagyságától. Ezért szükséges a forgalomterhelést több lépésben, a kapacitáskihasználtsági fok figyelembevételével végezni el. Visszatérve a forgalomszétosztásra, ennek két fő típusát lehet megkülönböztetni: a növekedési tényezős és a gravitációs modelleket.

I. *Növekedési tényezős modellek*

Ezeknél a modelleknél a kiindulási mintamátrix elemeit különböző növekedési tényezőkkel való szorzás révén úgy korrigáljuk, hogy az eredményül kapott mátrix sor- és oszlopösszegei az előírt értékkel egyezzenek meg. Az egyes módszerek a növekedési tényező képzési módjában különböznek egymástól. A módszerek jelentős része olyan, hogy az egymásutáni iterációkban felváltva teljes sorok és oszlopok egységes növekedési tényezővel kerülnek módosításra, s a növekedési tényező értéke csak az illető sor, illetve oszlop sorszámától függ.

Ez esetben az eljárás konvergens — mint ezt majd látjuk — és

$$h_{ij} = r_i g_{ij} s_j$$

alakú megoldás adódik, ahol az r_i, s_j tényezők az iterációkban alkalmazott növekedési szorzók szorzatának határértékei.

II. *Gravitációs modellek*

A gravitációs modell abból indul ki a mechanika NEWTON által megfogalmazott gravitációs törvényével analóg módon, hogy a két körzet közötti kölcsönhatás erőssége egyenesen arányos a körzetek tömegével, s fordítottan arányos a közöttük levő távolság valamilyen növekvő függvényével — az eredeti megfogalmazásban négyzetével —. Ennek az elvnek a forgalomszétosztásra való alkalmazásával, amikor a **H**

mátrix sor- és oszlopösszegei adottak, a következő alakú modellt szokták használni:

$$h_{ij} = r_i f(c_{ij}) s_j \tag{1.5}$$

$$r_i, s_j \geqq 0,$$

$$\sum_{j=1}^{q} h_{ij} = \bar{o}_i, \quad i = 1, 2, \ldots, p,$$

$$\sum_{i=1}^{p} h_{ij} = \bar{d}_j, \quad j = 1, 2, \ldots, q, \tag{1.6}$$

ahol $f(c)$ egy monoton csökkenő függvény, (ún. ellenállásfüggvény), c_{ij} pedig az i-edik és j-edik körzet közötti általánosított utazási egységköltség. Az $f(c)$ konkrét formájától függően különböző gravitációs modellek adódnak. Ezekkel a 3. részben foglalkozunk.

2. Növekedési tényezős modellek

2.1. *Feladat megfogalmazása*

A továbbiakban feltesszük, hogy ismert a $\mathbf{G}=(g_{ij})$ pxq méretű „minta" mátrix, az $\overline{\mathbf{O}}=(\bar{o}_i)$ $(i=1, 2, \ldots, p)$ és a $\overline{\mathbf{D}}=(\bar{d}_j)$, $(j=1, 2, \ldots, q)$ vektorok. A feladat annak a $\mathbf{H}=(h_{ij})$ mátrixnak a meghatározása, amelyre az (1.3), (1.4) feltételek teljesülnek és amely a „minta" mátrixra bizonyos értelemben legjobban „hasonlít".

A **H** mátrix meghatározására növekedési tényezős modelleket használtak először. Ezeknél feltesszük, hogy a jövőbeni utazásszám a jelenlegi utazásszám faktorokkal (ún. növekedési tényezőkkel) való felszorzásával adódik. Így a körzetek közötti utazási áramlatok arányairól feltételezzük implicite, hogy a jövőbeni hálózatban nem változnak a jelenlegihez képest.

Ezért a növekedési tényezős modelleket elsősorban közeli távlatú előrebecslésekre, vagy a jelenlegi forgalom szétosztására használhatjuk, mivel nem veszik figyelembe a hálózat strukturális változásait (KOLLER [18], MONIGL [19]). A **H** mátrixot a legtöbb módszernél iterációval határozzuk meg $\mathbf{H}^{(1)}, \mathbf{H}^{(2)}, \ldots$ mátrixokkal. Minden iterációban a $h_{ij}^{(k)}$ elemet a $h_{ij}^{(k-1)}$-ből egy $x_{ij}^{(k)}$-val való szorzással kapjuk meg. Azaz:

$$h_{ij}^{(1)} = x_{ij}^{(1)} g_{ij} \tag{2.1}$$

és

$$h_{ij}^{(k)} = x_{ij}^{(k)} h_{ij}^{(k-1)} = \prod_{m=1}^{k} x_{ij}^{(m)} g_{ij}.$$

A módszerek abban különböznek egymástól, hogy milyen módon kapjuk meg az $x_{ij}^{(k)}$ elemeket. A legtöbb esetben x_{ij} helyett egy, csak a sorra jellemző r_i és csak az oszlopra jellemző s_j-t keresünk, azaz azokat az

$$\mathbf{R} = (r_1, r_2, \ldots, r_p) \quad \text{és} \quad \mathbf{S} = (s_1, s_2, \ldots, s_q)$$

vektorokat, melyekre teljesülnek a következő feltételek:

$$\begin{aligned} r_i\left(\sum_{j=1}^{q} s_j g_{ij}\right) &= \bar{o}_i, \quad i = 1, 2, \ldots, p, \\ s_j\left(\sum_{i=1}^{p} r_i g_{ij}\right) &= \bar{d}_j, \quad j = 1, 2, \ldots, q. \end{aligned} \tag{2.2}$$

Ekkor **H** elemei a következőképp állnak elő:

$$h_{ij} = r_i g_{ij} s_j. \tag{2.3}$$

Mielőtt a modelleket ismertetnénk, foglalkozunk a növekedési tényezős módszerek egy jellemző tulajdonságával. Nevezetesen, mindegyik eljárásnál a $g_{ij}=0$ feltételből következik, hogy $h_{ij}=0$. Olyan i és j párok esetén, amelyeknél $\bar{o}_i$ és $\bar{d}_j$ egyaránt nagy, rendszerint nem jogos a $h_{ij}=0$ eredmény. A $g_{ij}=0$ lehetőség viszont előfordul olyan esetekben is, ha g_{ij} kicsi, és a mérési módszerek pontatlansága miatt nullának adódik.

A nulla elemek korrigálására két eljárást mutatunk be. Mindkettő a mért **G** és a számított **E** mátrix elemeinek lineáris kombinációjaként állítja elő a $\bar{\mathbf{G}}=(\bar{g}_{ij})$ korrigált mátrixot.

Az első eljárásnál csak az **O**, **D** vektorok és a **G** mátrix ismert, e_{ij}-t ezekből számoljuk:

$$e_{ij} = \frac{o_i d_j}{w}, \tag{2.4}$$

ahol w a g_{ij} elemek összege (KIRBY—LEESE [16]). Az e_{ij} elem nagy, ha o_i és d_j mindegyik nagy és kicsi, ha valamelyik, vagy mindkettő kicsi.

E elemeinek összege éppen w. Ha a forrás-nyelőpontok közötti utazási költségek $\mathbf{C}=(c_{ij})$ költségmátrixa is ismert, e_{ij} előállítására a következő módszer javasolható (KIRBY [15]):

$$e_{ij} = \frac{Kc_{ij}\left[\sum_i \frac{g_{ij}}{c_{ij}}\right]\left[\sum_j \frac{g_{ij}}{c_{ij}}\right]}{\sum_{i,j} \frac{g_{ij}}{c_{ij}}}, \tag{2.5}$$

ahol a K faktor biztosítja, hogy $\sum_{i,j} e_{ij}=w$. A (2.5) szerint az e_{ij} érték egyenesen arányos a költséggel, a forráspontból kimenő forgalomra jutó összköltséggel és a nyelőpontba befutó forgalomra jutó önköltséggel és fordítottan arányos az egy utazásra eső átlagköltséggel. Az előre megadott vagy a fenti módon számított **E** mátrixból meghatározunk két további faktort, az ún. koncentrációs paramétert (k) és a súlyozási tényezőt (z):

$$k = \frac{\left(w^2 - \sum_{i,j} g_{ij}^2\right)}{\sum_{i,j} (g_{ij} - e_{ij})^2} \tag{2.6}$$

és

$$z = \frac{w}{w+k}.$$

Ezek után a $\bar{\mathbf{G}}=(\bar{g}_{ij})$ mátrixot az alábbiakban határozzuk meg:

$$\bar{g}_{ij} = zg_{ij}+(1-z)e_{ij}. \tag{2.7}$$

Ez a módszer arra is alkalmas, hogy két honnan-hová mátrix esetén a megfelelően módosított kiindulási mátrixot meghatározzuk a (2.7) felhasználásával.

KIRBY—LEESE [16] a fenti eljárást az 1967-es „SELNEC" tanulmány adatai alapján mutatja be. A honnan-hová mátrix eredetileg 28 nulla elemet tartalmazott. A megfigyelt és a modellmátrix kombinációja segítségével 12 nulla elemet tudott eltüntetni.

Megjegyezzük, hogy ezeket a módszereket a gravitációs típusú modellek esetén is alkalmazhatjuk.

A továbbiakban összefoglaljuk a modelleket, részletesebb leírást az [1] dolgozat tartalmaz.

2.2. *Fontosabb modellek*

Globális tényezős modell

A globális tényezős modell egyetlen szorzótényezőt használ az összes körzetre. A jövőbeni forgalmat a jelenlegi forgalom globális faktorral való megszorzásával kapjuk. Az univerzális tényezőt és az előrebecsült forgalmat a következőképp számítjuk:

$$h_{ij} = \frac{\bar{w}}{w} g_{ij}, \quad i = 1, 2, \ldots, p, \quad j = 1, 2, \ldots, q, \tag{2.8}$$

ahol

$$\bar{w} = \sum_{i=1}^{p} \bar{o}_i, \quad w = \sum_{i=1}^{p} o_i.$$

Azonnal látszik, hogy a (2.8) eljárás az (1.3), (1.4) feltételeket nem elégíti ki.

A fenti probléma kiküszöbölésére fejlesztették a módszert tovább. Ilyen továbbfejlesztett módszereket ismertetünk a továbbiakban.

Átlagtényezős modell

Az átlagtényezős modellben minden egyes körzethez meghatározunk egy növekedési tényezőt. Az i-edik körzetből a j-edik körzetbe menő forgalom meghatározása ezek után az i-edik és j-edik körzethez tartozó tényező számtani átlagának felhasználásával történik. Az i-edik körzethez tartozó tényezőt az $\bar{o}_i/o_i$ hányados adja

meg. A h_{ij} elem meghatározását a következőképp végezzük:

$$h_{ij} = \frac{1}{2} g_{ij} \left(\frac{\bar{o}_i}{o_i} + \frac{\bar{d}_j}{d_j} \right). \tag{2.9}$$

A (2.9) eredményeképp kapott **H** mátrixra azonban az (1.3), (1.4) feltételek nem teljesülnek. Ezért célszerű a (2.9) eljárás helyett az alábbi iterációt alkalmazni (EVANS [9]).

$$\begin{aligned} &0.\ \text{lépés:} \quad h_{ij}^{0} = \frac{1}{2} g_{ij} \left[\frac{\bar{o}_i}{o_i} + \frac{\bar{d}_j}{d_j} \right], \\ &k+1.\ \text{lépés:} \quad h_{ij}^{(k+1)} = \frac{1}{2} h_{ij}^{(k)} \left[\frac{\bar{o}_i}{o_i^{(k)}} + \frac{\bar{d}_j}{d_j^{(k)}} \right], \end{aligned} \tag{2.10}$$

ahol

$$o_i^{(k)} = \sum_{j=1}^{q} h_{ij}^{(k)}, \quad d_j^{(k)} = \sum_{i=1}^{p} k_{ij}^{(k)}, \quad i = 1, 2, \ldots, p, \quad j = 1, 2, \ldots, q.$$

A fenti eljárásban az aktuális $\mathbf{H}^{(k)} = (h_{ij}^{(k)})$ mátrix elemeit tovább módosítjuk az előre lefixált $\bar{o}_i$ és a k-adik iterációban kapott $o_i^{(k)}$ elemek, valamint a $\bar{d}_j$ és $d_j^{(k)}$ elemek felhasználásával. Így az eljárás mindinkább közeledik az $\bar{o}_i / o_i^{(k)} \sim 1$, illetve a $\bar{d}_j / d_j^{(k)} \approx 1$ feltételekhez ($i = 1, 2, \ldots, p$, $j = 1, 2, \ldots, q$). Megállási kritériumként az 1—5%-os eltérés elérését alkalmazzák.

Fratar modell

Az eredeti modellt FRATAR 1954-ben *Clevelandban (Ohio)* alkalmazta és ennek a tapasztalatait publikálta (FRATAR [11]). Módszere az első két módszernek továbbfejlesztett változata, amelynek azonban a számítási igénye is megnövekedett. Az i-edik körzetből a j-edik körzetbe menő jövőbeni forgalmat a jelenlegi forgalomból számítjuk, de figyelembe vesszük az i-edik körzetből kimenő és a j-edikbe befutó forgalmat is. Minden körzetre meghatározzuk a kimenő és bemenő forgalomra jellemző növekedési faktort. Az i-edikből kimenő forgalomra jellemző tényezőt az alábbi összefüggés adja meg:

$$r_i = \frac{\sum_{i=1}^{q} g_{ij}}{\sum_{j=1}^{q} d_j g_{ij}} = \frac{o_i}{\sum_{j=1}^{q} d_j g_{ij}}. \tag{2.11}$$

Hasonló tényező adható meg a j-edik körzetbe beáramló forgalomra is:

$$s_j = \frac{\sum_{i=1}^{p} g_{ij}}{\sum_{i=1}^{p} o_i g_{ij}} = \frac{d_j}{\sum_{i=1}^{p} o_i g_{ij}}. \tag{2.12}$$

Ezek után azt a feltételezést tesszük, hogy az i-edik körzetből a j-edikbe menő előrebecsült forgalom egyenesen arányos az eredeti forgalommal, az i-edik körzetből kiáramló és a j-edikbe befolyó forgalom növekedési arányával és az i-hez és j-hez tartozó tényezők átlagával, azaz

$$h_{ij} = \frac{1}{2} g_{ij} \frac{\bar{o}_i}{o_i} \frac{\bar{d}_j}{d_j} (r_i + s_j). \tag{2.13}$$

Az (2.11)—(2.13) képletek eredményeképp kapott eljárásra az (1.3), (1.4) feltételek nem teljesülnek, ezért az alábbi iterációs eljárást alkalmazzuk:

$$\text{0. lépés:} \quad h_{ij}^{(0)} = \frac{1}{2} g_{ij} \frac{\bar{o}_i}{o_i} \frac{\bar{d}_j}{d_j} (r_i + s_j),$$

$$k+1\text{-edik lépés:} \quad h_{ij}^{(k+1)} = \frac{1}{2} h_{ij}^{(k)} \frac{\bar{o}_i}{o_i^{(k)}} \frac{\bar{d}_j}{d_j^{(k)}} (r_i^{(k)} + s_j^{(k)}), \tag{2.14}$$

ahol

$$o_i^{(k)} = \sum_{j=1}^{q} h_{ij}^{(k)}, \; d_j^{(k)} = \sum_{i=1}^{p} h_{ij}^{(k)}$$

és

$$p_j^{(k)} = \frac{\bar{d}_j}{d_j^{(k)}}, \; q_i^{(k)} = \frac{\bar{o}_i}{o_i^{(k)}}$$

$$r_i^{(k)} = \frac{o_i^{(k)}}{\sum_{j=1}^{q} p_j^{(k)} h_{ij}^{(k)}}, \quad s_j^{(k)} = \frac{d_j^{(k)}}{\sum_{i=1}^{p} q_i^{(k)} h_{ij}^{(k)}}. \tag{2.15}$$

Megállási kritériumként ismét használhatjuk az $\bar{o}_i/o_i^{(k)} \approx 1$, illetve a $\bar{d}_j/d_j^{(k)} \approx 1$, $i=1, 2, \ldots, p$, $j=1, 2, \ldots, q$ feltételeket. Gyakorlatban jól bevált az $|r_i^{(k+1)} - r_i^{(k)}|$, illetve $|s_j^{(k+1)} - s_j^{(k)}|$ értékek vagy a $\max_{i,j} |h_{ij}^{(k+1)} - h_{ij}^{(k)}|$ érték kis küszöb alatt maradásának ellenőrzése is.

Detroiti modell

A módszert a detroiti városi forgalomszétosztásra használták először (BEVIS [3]). A modell megkísérelte a *Fratar módszer* előnyeit átvéve egyszerűsíteni a bonyolult, sok számolást igénylő eljárást.

A *Fratar modellben* levő, az i-edik körzetből kiáramló forgalomra jellemző r_i és a j-edikbe bemenő forgalomra jellemző s_j faktorok helyett összesen egy faktort alkalmazott:

$$F = \frac{\bar{w}}{w}. \tag{2.16}$$

A feladatot az alábbi iterációval oldja meg:

$$\text{0. lépés:} \quad h_{ij}^{(0)} = g_{ij}\frac{q_i \cdot p_i}{F},$$

(2.17)

$$k+1.\ \text{lépés:} \quad h_{ij}^{(k+1)} = h_{ij}^{(k)}\frac{q_i^{(k)} \cdot p_j^{(k)}}{F^{(k)}} \quad k = 1, 2, \ldots,$$

ahol

$$F^{(k)} = \frac{\sum_{i=1}^{p} \bar{o}_i}{\sum_{i=1}^{p} o_i^{(k)}}, \quad q_i^{(k)} \quad \text{és} \quad p_j^{(k)}$$

pedig a *Fratar modellben* megadott értékek.

Hogy az (1.3), (1.4) feltételeket kielégítsük, további kiegyenlítő számítások szükségesek.

Furness módszere

Az előző algoritmusokban minden lépésben új növekedési tényezőket határoztunk meg. A *Furness eljárás* a sor- és oszlop vektorokkal végez műveleteket addig, míg az eljárás a megoldáshoz nem konvergál.

Az iterációt az alábbi képletekkel adjuk meg:

$$\text{1. lépés:} \quad h_{ij}^{(0)} = g_{ij},\ h_{ij}^{(1)} = h_{ij}^{(0)}\frac{\bar{o}_i}{\sum_{k=1}^{p} g_{ik}},$$

közbülső lépés: ($k = 2m$ alakú)

$$h_{ij}^{(k)} = \frac{\bar{d}_j}{\sum_{s=1}^{p} h_{sj}^{(k-1)}} h_{ij}^{(k-1)}, \quad h_{ij}^{(k+1)} = \frac{\bar{o}_i}{\sum_{s=1}^{q} h_{is}^{(k)}} h_{ij}^{(k)}. \tag{2.18}$$

Az iteráció akkor ér véget, ha a fenti (2.18) képletben a törtek értéke elég közel van egyhez. Meg lehet mutatni, hogy a fenti $h_{ij}^{(0)}$ helyett tetszés szerinti pozitív számsorból kiindulva ugyanazt a megoldást kapjuk. A fenti eljárásban a 0. lépésben a sorösszegekből indultunk ki. Az iteratív eljárásban páratlan esetben a $h_{ij}^{(k+1)}$ mátrix sorainak összege $\bar{o}_i$-vel, páros esetben a $h_{ij}^{(k)}$ mátrix oszlopainak összege $\bar{d}_j$-vel egyezik meg. Ezért

$$\sum_{i,j} h_{ij}^{(k)} = \sum_{i=1}^{p} \bar{o}_i = \sum_{j=1}^{q} \bar{d}_j = \bar{w}. \tag{2.19}$$

Páratlan esetben a szorzófaktor — jelöljük r_i-vel — a következő:

$$r_i^{(k+1)} = \frac{\bar{o}_i}{\sum_{j=1}^{q} h_{ij}^{(k)}}. \tag{2.20}$$

Páros esetben a megfelelő $s_j^{(k)}$ szorzófaktor:

$$s_j^{(k)} = \frac{\partial_j}{\sum_{i=1}^{p} h_{ij}^{(k-1)}}. \tag{2.21}$$

Ezért a megfelelő $h_{ij}^{(k+1)}$ és $h_{ij}^{(k)}$ elemek a következőképp néznek ki:

$$\begin{aligned} h_{ij}^{(k)} &= s_j^{(k)} h_{ij}^{(k-1)}, \\ h_{ij}^{(k+1)} &= r_i^{(k+1)} h_{ij}^{(k)}. \end{aligned} \tag{2.22}$$

2.3. *A modell alkalmazási lehetősége*

EVANS [9], HUTCHINSON [12], KOLLER [18], MONIGL [19] szerint a növekedési tényezős modellek nem alkalmasak hosszútávú előrebecslésre. Ennek az az oka, hogy a hálózat struktúrájában bekövetkezendő változásokat nem követik ezek a módszerek, és a jelenlegi forgalmi szituációkat használják.

A növekedési tényezős módszereket a tapasztalatok szerint a következő esetekben célszerű használni:

a) olyan városok esetén, ahol a városszerkezet a jövőben nem változik alapvetően (HUTCHINSON [12]);
b) rövid távú előrebecslések esetén (legfeljebb 5 év);
c) a más módszerrel korábban előrebecsült forgalmi mátrixok naprakész állapotra hozásakor;
d) gravitációs és versengő lehetőségek módszernél kiegyenlítő számításokra (adott sor- és oszlopösszegek);
e) olyan közúti modellek esetén, amelyekben hálózatmódosítás nincs betervezve.

A jelenlegi metodológia olyan esetben használja a modelleket, amikor a forrás-nyelő pont kívül esik a tanulmányozandó területen. Ilyenkor a megfelelő adatok csak arra a területre vannak meg, amelyet vizsgálunk. Fontos viszont a környező területtel való jövőbeni forgalmi viszonyok tanulmányozása is, ami néhány adat ismeretében egyszerű módszerrel lehetséges. (L. STOPHLER—MEYBUR [21], 127. o.)

2.4. *Megoldhatósági kérdések*

Az eddigiekben hallgatólagosan feltettük, hogy a kitűzött feladatok megoldhatók, azaz létezik olyan megoldás, amelyre a sor- és oszlopösszegekre előírt feltételek teljesülnek. További fontos kérdés, hogy az iterációs eljárások konvergensek-e és ha igen, mindegyik ugyanazt az eredményt szolgáltatja-e. Nevezzük a továbbiakban az olyan megoldást lehetségesnek, amely kielégíti az (1.3, 1.4) feltételeket. Az könnyen belátható, hogy lehetséges megoldás nem mindig létezik.

Tekintsük például a következő 3 mátrixot a megadott sor- és oszlopösszegekkel:

$$\mathbf{A} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & 0 \end{bmatrix}\begin{matrix}2\\8\end{matrix}, \quad \begin{matrix}7 & 3\end{matrix} \qquad \mathbf{B} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & 0 \end{bmatrix}\begin{matrix}3\\7\end{matrix}, \quad \begin{matrix}7 & 3\end{matrix} \qquad \mathbf{C} = \begin{bmatrix} 1 & 3 \\ 6 & 0 \end{bmatrix}\begin{matrix}4\\6\end{matrix}, \quad \begin{matrix}7 & 3\end{matrix}$$

Az **A** mátrix esetében a feladatnak nincs lehetséges megoldása. A **B** mátrix esetében megoldás nem áll elő $r_i g_{ij} s_j$ alakban. A **C** mátrix esetében a $h_{11}=1$, $h_{12}=3$, $h_{21}=6$, $h_{22}=0$ lehetséges megoldás, és (2.3) alakban áll elő, ha $r_1=1$, $r_2=6$, $s_1=1$, $s_2=3$. A feladat megoldhatóságát eldönthetjük a *Kőnig—Hall-tétel* felhasználásával. Legyen a $\mathbf{G}=(g_{ij})$ mátrix egy kvalifikációs mátrix. A g_{ij} elem most azt jelenti, hogy ha $g_{ij}>0$, akkor az i-edik csoportban levő o_i számú munkás mindegyike ért a j-edik munkahelyen levő d_j számú munkához. Legyen $I=(1, 2, \ldots, p)$ és a $J=(1, 2, \ldots, q)$ és jelöljük $T \subset I$ indexhalmaz esetén a **G** mátrix által kvalifikált $j \in J$ indexek halmazát $J(T)$-vel.

A megoldás létezésének szükséges és elegendő feltétele, hogy bármely $T \subset I$ indexhalmazra és $J(T)$-re fennálljon az alábbi összefüggés:

$$\sum_{i \in T} \bar{o}_i \leqq \sum_{j \in J(T)} \bar{d}_j .$$

A globális tényezős eljárás egy képlet kiszámításából áll, így a konvergencia kérdése nem kerül szóba.

A *Furness eljárás* konvergenciájának bizonyítását először BREGMAN [4] közölte. Azóta számos szerző adott bizonyítást különböző eszközök felhasználásával, így EVANS [9] $g_{ij}>0$ esetre bizonyította. Nála egyben az is bizonyításra került, hogyha $g_{ij}>0$ minden i és j esetén, akkor a

$$h_{ij} = r_i g_{ij} s_j$$

$$\sum_j h_{ij} = \bar{o}_i$$

$$\sum_i h_{ij} = \bar{d}_j$$

feladat megoldása a h_{ij}-re egyértelmű. KLAFSZKY [17] geometriai programozást, CSISZÁR [8] az információ divergencia geometriai tulajdonságait használta a bizonyításhoz. Az eljárás konvergenciájának gyakorlati tapasztalatairól számolnak be D'ESOPO—LEFKOWITZ [6] dolgozatukban.

Ha két mátrixból kiindulva ugyanazt a végeredményt kapjuk a *Furness eljárással*, akkor a két mátrix egy közbülső lépésben áttranszformálható egymásba. Ezt mutatjuk meg a következőkben.

Legyen **U** egy pozitív mátrix, amelyből a megadott $\bar{o}_1, \bar{o}_2, \ldots, \bar{o}_p$; $\bar{d}_1, \bar{d}_2, \ldots, \bar{d}_q$ sor- és oszlopösszegekkel a *Furness eljárással* előállítjuk a **V** mátrixot.

2.1. TÉTEL. $\mathbf{H}=\mathbf{V}$ akkor és csakis akkor, ha vannak olyan $a_1, a_2, \ldots, a_p$, $b_1, b_2, \ldots, b_q$ pozitív számok, amelyekre $u_{ij}=a_i g_{ij} b_j$.

Bizonyítás: Tegyük fel, hogy $\mathbf{H}=\mathbf{V}$. Megmutatjuk, hogy **U** elemei előállnak a fenti alakban. Az eljárás konvergenciája miatt vannak olyan $r_1, r_2, \ldots, r_p$,

$s_1, s_2, \ldots, s_q$, $R_1, R_2, \ldots, R_p$ és $S_1, S_2, \ldots, S_q$ pozitív számok, amelyekre:

$$h_{ij} = r_i g_{ij} s_j,$$

$$v_{ij} = R_i u_{ij} S_j.$$

A feltételezés miatt $r_i g_{ij} s_j = R_i u_{ij} S_j$, innen u_{ij}-t kifejezve

$$u_{ij} = \frac{r_i}{R_i} \frac{s_j}{S_j} g_{ij} = a_i g_{ij} b_j.$$

Most tegyük fel, hogy $u_{ij} = a_i g_{ij} b_j$ alakú. A $\mathbf{V} = (v_{ij})$ egyértelmű megoldás

$$v_{ij} = R_i u_{ij} S_j$$

alakú és kielégíti a sor- és oszlopösszegekre vonatkozó feltételeket.

A $\mathbf{H}$ mátrix szintén kielégíti ezeket a feltételeket, és

$$h_{ij} = r_i g_{ij} s_j.$$

De a feltételezés miatt $g_{ij} = \frac{u_{ij}}{a_i b_j}$, ezt beírva a fenti egyenlőségbe kapjuk:

$$h_{ij} = \frac{r_i}{a_i} u_{ij} \frac{s_j}{b_j},$$

azaz h_{ij} ugyanolyan alakban áll elő, mint v_{ij}, így szükségképpen $h_{ij} = v_{ij}$ ($\mathbf{H} = \mathbf{V}$).

A *Detroiti eljárás* szintén ilyen alakban szolgáltatja a megoldást, ezért a két eljárás végeredménye megegyezik. Azt is könnyen megmutathatjuk, hogy az *átlagtényezős* és *Fratar eljárás* eredménye rendszerint nem egyezik meg a *Furness* és a *Detroiti módszer* eredményével. A *Fratar* és az *átlagtényezős módszernél* ugyanis a módosító tényezők között nem szorzás jel szerepel, azaz

$$h_{ij} = [a_i + b_i] g_{ij}$$

alakú. Így a két különböző (additív és multiplikatív) eljáráscsoport eredménye akkor egyezik meg, ha

$$a_i + b_j = r_i s_j$$

feltétel teljesül. Gyakorlati tapasztalataink és megegyező irodalmi közlések (lásd például MORRIS [20]) szerint a *Fratar* és az *átlagtényezős eljárások* is konvergensek

2.5. *Növekedési tényezős módszerek összehasonlítása*

A növekedési tényezős módszerek számítógépes vizsgálatához elkészítettük a fejezetben leírt módszerek programját PL/I nyelven.

A növekedési tényezős módszerek szubrutinjaihoz szükséges számítógépes erőforrás igényeket az 1. táblázatban foglaljuk össze.

A memóriaigényt tekintve gyakorlatilag minden eljárásnak azonos a helyigénye. Az eljáráshoz szükséges utasítások számát tekintve a globális tényezős módszerhez a legkevesebb, a többi eljáráshoz gyakorlatilag egyforma számú utasítás szükséges.

1. TÁBLÁZAT

Növekedési tényezős eljárások számítógépes programjának jellemző adatai

Módszer	Memóriaigény	Utasítások száma	Műveletek száma iterációnként		
			+/−	×	:
Globális	$2n(n+1)$	35	$n(n+1)$	n^2	n
Átlagtényezős	$2n(n+1)$	115	$3n^2$	$3n^2$	$2n$
Fratar	$2n(n+3)$	128	$4n^2$	$6n^2$	$4n$
Detroiti	$2n(n+3)$	111	$2n^2$	$3n^2$	$n(n+3)$
Furness	$2n^2+3n$	125	n^2	n^2	n^2

Az eljárások programjai PL/I-ben készültek, s így a megadott utasításszám is erre a nyelvre vonatkozik. A 3—5. oszlopokban adtuk meg az egy iterációhoz szükséges összeadások/kivonások, szorzások és osztások számát. E szerint az iterációs eljárások közül az egy iterációhoz szükséges legkevesebb időt az átlagtényezős módszer igényli.

A módszereket a budapesti 7 körzetes tömegközlekedési (lakás—munkahely) mátrixokon ellenőriztük. A budapesti kiinduló mátrixot az 1. melléklet tartalmazza. Mivel az egy iterációhoz szükséges lépések száma önmagában nem sokat mond, megmértük a különböző eljárásokhoz szükséges teljes futási időt. A 2. táblázat első oszlopa a módszerek konvergenciájához szükséges iterációs lépések számát tartalmazza.

Leállási kritériumként a megadott $\bar{o}_i$ sor- és $\bar{d}_i$ oszlopösszegek és a $\mathbf{H}^{(k)}=(h_{ij}^{(k)})$ mátrix sor- és oszlopösszegeinek maximális hányadosát adtuk meg.

Az eljárás befejeződik, ha a

$$0{,}95 \leqq \max\left(\max_i \frac{\bar{o}_i}{o_i^{(k)}},\ \max_j \frac{\bar{d}_j}{d_j^{(k)}}\right) \leqq 1{,}05 \tag{2.23}$$

feltétel teljesül.

A *Fratar* és a *Furness módszernél* két, a *Detroiti módszernél* pedig 3 iterációs lépésre volt szükség a konvergenciához. Az átlagtényezős módszer 20 iteráció után is csak 0,72 hányadost adott.

Az egy iterációhoz szükséges idő gyakorlatilag megegyezik minden módszernél (2. táblázat 2. oszlopa). Az összes szükséges iterációs lépések száma a *Fratar* és

2. TÁBLÁZAT

Konvergencia ellenőrző adatok a budapesti 7×7-es mátrixra

Módszer	Iterációk száma	Futási idők		Maximális		D
		Egy iteráció	Összes	gyök	hányados	
Globális	1	0,11	0,11	117,8	1,82	
Átlagtényezős	20	0,12	24,0	86,4	1,72	3,77
Fratár	2	0,13	0,27	2,81	1,02	1,64
Detroiti	3	0,13	0,41	5,06	0,97	2,06
Furness	2	0,13	0,26	6,64	0,98	2,07

Furness eljárásnál megegyezik, a *Detroiti* pedig egy harmad résszel több számítási időt vett igénybe (3. oszlop). Az iterációs eljárások pontosságának mérésére a (2.23) mellett két további mérőszámot is meghatároztunk. Az egyik a sor- és oszlopösszegek egymástól való távolsága, azaz az

$$\varepsilon_1 = \sqrt{\sum_i (\bar{o}_i - o_i^{(k)})^2},$$

(2.24)

$$\text{illetve} \quad \varepsilon_2 = \sqrt{\sum_j (\bar{d}_j - d_j^{(k)})^2}$$

érték volt. A 4. oszlopba kiírtuk az $\varepsilon = \max(\varepsilon_1, \varepsilon_2)$ értékeket. Ezt tekintve a *Fratar módszer* bizonyult a legjobbnak, ezt követte a *Detroiti* és a *Furness* közel azonos értékkel, az *átlagtényezős módszer* a legpontatlanabb — végül a globális tényezős módszer messze rosszabb eredményt szolgáltatott.

A (2.24) hányadosok maximumát az 5. oszlop tartalmazza. E szerint csak a három utolsó módszer ad elfogadható eredményt, az első kettő pedig igen pontatlan. A harmadik mérőszám az eredeti és az előrebecsült tábla egymáshoz való viszonyát van hivatva megmutatni. Erre az alábbi összefüggést használtuk:

$$D = \frac{\sum_{i,j} (g_{ij} - h_{ij})^2}{\sum_{i,j} g_{ij}}. \tag{2.25}$$

A legpontosabb illeszkedést e tekintetben a *Fratar módszer* mutatta; közel azonos eredményt szolgáltatott a *Detroiti* és *Furness módszer*, végül az *átlagtényezős módszer* adta a legmagasabb értéket.

A részletes futási eredményeket a 2. mellékletben adtuk meg. Mindegyik módszer esetén közöljük a **H** mátrix elemeit, az $o_i^{(k)}$ és $d_j^{(k)}$ értékeket, elemenként az $\bar{o}_i/o_i^{(k)}$ és $\bar{d}_j/d_j^{(k)}$ hányadosokat és a (2.23)—(2.25) értékeket. A táblázatok elemenkénti és módszerenkénti összehasonlítása meglehetősen nehézkes. Ezért kiválasztottuk a mindegyik módszernél eléggé pontos oszlopösszeget szolgáltató 6. oszlopot. A különböző módszerekhez tartozó 6. oszlop elemet a 3. táblázatban foglaltuk össze.

Látható, hogy a módszerek ugyanarra az elemre igen eltérő eredményeket adnak. A *globális* és *átlagtényezős módszer* az eredeti **G** mátrix elemeihez közelebb álló eredményeket szolgáltatott. A *Fratar, Detroiti* és *Furness módszerek* eredményei viszont csak kevéssé térnek el egymástól. Az eltérés az 1, 2, 4, 5, 6, 7. elemeket tekintve 1—2%-os, a 3. elemnél is csak 4%-os. Az 1. és 2. módszert összevetve az utóbbiak-

3. TÁBLÁZAT

A **H** mátrix 6. oszlopának elemei különböző eljárásoknál

Módszer	1	2	3	4	5	6	7
Globális	8,4	10,2	7,4	2,8	14,9	50,3	21,4
Átlagtényezős	9,5	10,6	8,6	3,7	15,3	52,6	20,2
Fratar	10,5	9,9	9,6	5,4	13,8	49,1	15,3
Detroiti	10,4	9,7	9,6	5,5	13,6	48,1	15,1
Furness	10,5	9,8	10,0	5,4	13,7	49,4	15,2
Kiindulás	9,0	11,0	8,0	3,0	16,0	54,0	23,0

kal, 25% körül relatív eltérést is tapasztalhatunk. Hasonló összefüggést kapunk az egyéb sorok-oszlopok összevetésével is.

BROKKE és MERTZ [5] a *Fratar*, és a *Detroiti módszereket* vetette össze *Washington* esetén, BEVIS [3] pedig *Detroit* esetén vizsgálta ugyanezen módszereket. Eredményül azt kapták, hogy a vizsgált módszerek gyakorlatilag ugyanazt az eredményt szolgáltatják. MORRIS [20] az *átlagtényezős*, a *Furness*, a *Fratar* és a *Detroiti módszereket* vizsgálta dolgozatában. Szerinte — a mi megállapításunkkal egyezően — az utóbbi három módszer közel azonos eredményt szolgáltat, az *átlagtényezős módszer* eredménye viszont eltér ettől. A konvergencia időt tekintve a *Fratar eljárást* mutatja ki leggyorsabbnak, szemben a mi futási tapasztalatainkkal.

Megjegyzendő, hogy ő csak becsült időket vett figyelembe szemben a mi mért eredményeinkkel.

3. Gravitációs modell

3.1. *A feladat megfogalmazása*

Még a múlt század végén állapították meg azt az összefüggést *Németországban* konkrét forgalmi adatok alapján, hogy két, vasútvonallal ellátott város közötti forgalom fordítottan arányos a városok közötti távolsággal (illetve inkább annak négyzetével), s egyenesen arányos a két város „nagyságával" — ami pl. a lakosság számával, vagy a foglalkoztatottak számával mérhető. Képletben kifejezve:

$$h_{ij} = k\frac{r_i s_j}{c_{ij}^2}. \tag{3.1}$$

Ez az összefüggés teljesen analóg a *Newton-féle gravitációs törvénnyel*. Innen származik az elnevezés: *gravitációs modell*. A későbbiek folyamán a távolság hatását kifejező $\frac{k}{c_{ij}^2}$ tényezőt az általános $f(d_{ij})$ „ellenállás"-függvénnyel — *„distant friction function"*, ill. *„Widerstandsfunktion"* — helyettesítették, melyről általában csak az van kikötve, hogy monoton csökkenő függvény. Tehát a (3.1) képlet így módosul:

$$h_{ij} = r_i f(c_{ij}) s_j. \tag{3.2}$$

Itt a c_{ij} már nemcsak távolságot, hanem utazási költséget, utazási időt is jelenthet — leggyakrabban ún. általánosított utazási költséget jelent, ami több tényezőből tevődik össze. A gravitációs modell alkalmazási köre az utóbbi 20 évben kibővült, az eredeti forgalomelőrebecslési feladat mellett a területi kölcsönhatások számos különböző fajtájának vizsgálatánál is felhasználják — az ún. regionális tudomány keretein belül.

A körzetek közötti forgalmi áramlatok meghatározására szolgáló gravitációs modellnek a körzetek kiinduló és beérkező forgalmára vonatkozó független becslések meglététől függően a következő típusait lehet megkülönböztetni:

— korlátozó feltétel nélküli modell, melynél a **H** mátrix sor-, ill. oszlopösszegeire nincs megkötés,
— kibocsátási oldalról korlátozott modell, melynél a sorösszegek adottak,

— forgalom-befogadó oldalról korlátozott modell, melynél az oszlopösszegek adottak,
— kétoldalról korlátozott modell, melynél mind a sor, mind az oszlopösszegek adottak.

Legtöbbször a kétoldalról korlátozott gravitációs modellt használják, ha az ellenkezőjét külön nem emeljük ki, akkor mi is mindig erre gondolunk.

Az ellenállásfüggvény számos különböző fajtája ismert. A leggyakrabban alkalmazott a negatív exponenciális függvény:

$$f_1(c) = e^{\alpha c} \qquad (\alpha < 0).$$

További egyszerű és gyakori függvénytípusok:

$$f_2(c) = c^{\beta} \qquad (\beta < 0),$$

$$f_3(c) = e^{\alpha c} c^{\beta} \quad (\alpha, \beta < 0).$$

Egyéb bonyolultabb, és esetleg több paramétert alkalmazó függvénytípusokra itt nem térünk ki. Az ellenállásfüggvényben szereplő paraméter (α, β ill. α és β) meghatározása a modell ún. kalibrálásának feladata.

A (kétoldalról korlátozott) gravitációs modell a következő feladat megoldását jelenti:

keresendők olyan $r_i, s_j > 0$ értékek, melyekre teljesül, hogy a

$$h_{ij} = r_i f(c_{ij}) s_j$$

értékek a mátrix sor- és oszlopösszegeire előírt feltételeket teljesítik:

$$\sum_{j=1}^{q} h_{ij} = \bar{o}_i, \quad i = 1, \ldots, p,$$

$$\sum_{i=1}^{p} h_{ij} = \bar{d}_j, \quad j = 1, \ldots, q,$$

ahol $f(c_{ij}) > 0$ $(i = 1, \ldots, p;\ j = 1, \ldots, q)$ megadott értékek.

Ismeretes (bizonyított), hogy ennek a feladatnak létezik megoldása, s a megoldás egyértelmű a h_{ij} értékekre — a *Furness-féle iterációs módszer*, s más hasonló, multiplikatív módszerek ehhez a megoldáshoz konvergálnak. Tehát, ha az ellenállásfüggvény paraméterét (vagy paramétereit) meghatározzuk, akkor ettől kezdve már a gravitációs modell megoldása egyértelmű (és a tapasztalatok szerint még elég nagy méretek mellett is számítástechnikailag is viszonylag könnyen kivitelezhető).

A továbbiakban elsősorban az exponenciális ellenállásfüggvényű gravitációs modell kalibrálásával, s ehhez kapcsolódó kérdésekkel fogunk foglalkozni.

3.2. *Három modellvariáns ekvivalenciájának bizonyítása a geometriai programozás segítségével*

A (kétoldalról korlátozott) gravitációs modellt exponenciális ellenállásfüggvény mellett fogjuk tanulmányozni a következőkben, mely így fogalmazható meg: keresendők olyan α és $r_i, s_j > 0$ $(i=1, \ldots, p, j=1, \ldots, q)$ értékek, melyekre:

$$\left.\begin{aligned} &h_{ij} = r_i \exp(\alpha c_{ij}) s_j \text{ mellett} \\ &\sum_{j=1}^{q} h_{ij} = \bar{o}_i, \quad i = 1, \ldots, p, \\ &\sum_{i=1}^{p} h_{ij} = \bar{d}_j, \quad j = 1, \ldots, q. \end{aligned}\right\} (G)$$

Mint láttuk, ennek a feladatnak adott α esetén egyértelmű a megoldása a h_{ij} értékekre — az r_i, s_j értékekre csak egy konstans szorzó erejéig egyértelmű a megoldás, mivel, ha r_i $(i=1, \ldots, p)$, s_j $(j=1, \ldots, q)$ megoldás, akkor $\bar{r}_i = r_i k$ $(i=1, \ldots, p)$, $\bar{s}_j = s_j/k$ $(j=1, \ldots, q)$ is nyilván megoldás. Az α értékének rögzítéséhez azonban szükséges még egy pótlólagos feltétel, a kalibrációs kritérium megadása. A szakirodalom tanúsága szerint legismertebb ilyen kritérium az ún. „összköltség feltétel".

A. WILSON-tól származik ez a feltétel (l. pl. [23]-ban), amely a **H** utazási mátrixhoz tartozó

$$C = \sum_{i=1}^{p} \sum_{j=1}^{q} c_{ij} h_{ij}$$

„utazási összköltség" értékét rögzíti előzetesen, ami az α értékét már egyértelműen meghatározza, mint az bizonyítható. A C utazási összköltség jövőben várható értékét a jelenlegi állapothoz — ill. az alapul szolgáló forgalomfelvétel időpontjához, az ún. diagnózis-állapothoz — tartozó

$$C(g) = \sum_{i=1}^{p} \sum_{j=1}^{p} \bar{c}_{ij} g_{ij}$$

utazási összköltségből kiindulva lehet becsülni (a jövőben várható c_{ij} fajlagos utazási költségek általában különböznek a diagnózis állapothoz tartozó $\bar{c}_{ij}$ értékektől). Tehát az összköltség feltételes gravitációs modell a következő:

keresendők olyan α, valamint r_i, $s_j > 0$, h_{ij} értékek, melyekre:

$$h_{ij} = r_i \exp(\alpha c_{ij}) s_j$$

$$\sum_{j=1}^{q} h_{ij} = \bar{o}_i, \quad i = 1, \ldots, p,$$

$$\sum_{i=1}^{p} h_{ij} = \bar{d}_j, \quad j = 1, \ldots, q,$$

$$\sum_{i=1}^{p} \sum_{j=1}^{q} h_{ij} c_{ij} = C.$$

Most megmutatjuk, hogy ha ezt a modellt a jelenlegi („diagnózis") állapotra alkalmazzuk (tehát $\bar{o}_i$; $\bar{d}_j$-nek a g_{ij} elemekből álló mátrix sor-, ill. oszlopösszegeit, C-nek pedig $C(g)$-t, a c_{ij}-knek a $\bar{c}_{ij}$-ket vesszük), akkor olyan h_{ij} mátrixot kapunk, amely a lehető legkevésbé tér el a g_{ij} mátrixtól — ha a két mátrix eltérését az *„információ nyereség"* (más néven *„I-divergencia"*) mértékével mérjük. Tehát az összköltség-feltétel, mint kalibrációs elv egybe esik azzal a kézenfekvőbb elvvel, hogy olyan paraméterbecslési módszert keresünk, melyet a ténymátrixra alkalmazva, a kapott — gravitációs modellel szétosztott — mátrix lehető legjobban illeszkedik a ténymátrixra.

A bizonyítás a nemlineáris programozás egyik fajtája, a geometriai programozás dualitási tétele segítségével történik. Emellett itt járulékos eredményként megkapjuk még azt is, hogy az összköltség-feltételes gravitációs modell az összköltség-feltételes „entrópia-maximalizálási" modellel is ekvivalens — ez már korábban is ismert volt.

Most röviden megfogalmazzuk a geometriai programozási primál-duál feladatpárt és két dualitási tételt.

Primál feladat:

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

ahol

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

a_{ij} tetszőleges valós számok

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

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

Duál feladat:

$$(\mathrm{D})\begin{cases} \max v(\mathbf{y}) = \prod_{i=1}^{n} \left(\frac{C_i}{y_i}\right)^{y_i} \prod_{k=1}^{p} \lambda_k(\mathbf{y})^{\lambda_k(\mathbf{y})} \\ \text{ahol} \quad \lambda_k(\mathbf{y}) = \sum_{i \in J(k)} y_i \\ \text{feltéve, hogy} \\ \mathbf{y} = (y_1, \ldots, y_n) \geqq \mathbf{0} \\ \sum_{i \in J(0)} y_i = 1 \\ \sum_{i=1}^{n} a_{ij} y_i = 0, \quad j = 1, \ldots, m. \end{cases}$$

Gyakran a $v(\mathbf{y})$ duálfüggvény helyett annak logaritmusát, $\log v(\mathbf{y})$-t használják duál célfüggvénynek — ez a logaritmusfüggvény monotonitása miatt megtehető. A $c_i = \log C_i$ jelöléssel:

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

A (P) primál feladatot szuperkonzisztensnek nevezzük, ha van olyan $\mathbf{t} > \mathbf{0}$ megengedett megoldása, amely határozott egyenlőtlenséggel elégít ki minden korlátozó feltételt.

1. *(„Gyenge") dualitási tétel*

Tegyük fel, hogy a (P) feladat szuperkonzisztens és a $g_0(\mathbf{t})$ felveszi a primál megengedett halmazon az infimumát. Ekkor:

— a megfelelő (D) duál feladat konzisztens és a $v(\mathbf{y})$ felveszi a duál megengedett halmazon a supremumát,
— $\min_{\mathbf{t} \in (P)} g_0(t) = \max_{\mathbf{y} \in (D)} v(\mathbf{y})$,
— tetszőleges $(\bar{\mathbf{t}}, \bar{y})$ optimális megoldás-párra teljesülnek a következő, ún. egyensúlyi feltételek:

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

2. *(„Erős") dualitási tétel*

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

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

A geometriai programozás problémakörének részletesebb tárgyalása megtalálható pl. a [7], [13] munkákban.

Visszatérve a forgalomszétosztási problémára, a gravitációs modellel becsült h_{ij} ill. a megfigyelt g_{ij} utazási mátrix alapján definiálható egy „a-priori" és egy „a-posteriori" valószínűségeloszlás (annak a valószínűségére nézve, hogy egy véletlenül kiválasztott utazás éppen az i-edik körzetből a j-edik felé tart):

$$p_{ij} = \frac{h_{ij}}{\sum_{k,s} h_{ks}},$$

$$q_{ij} = \frac{g_{ij}}{\sum_{k,s} g_{ks}}.$$

Az információ-nyereséget, vagy másnéven *I-divergenciát* a két eloszlás között a következőképpen szokás definiálni:

$$I(g, h) = \sum_{i,j} q_{ij} \log \frac{q_{ij}}{p_{ij}}. \tag{3.3}$$

Ezt egyben tekinthetjük a h_{ij} és a g_{ij} mátrix közötti strukturális eltérés mérőszámának.

Vezessük be a következő jelöléseket:

$$\sum_{j=1}^{q} g_{ij} = a_i.$$

$$\sum_{i=1}^{p} g_{ij} = b_j.$$

$$\sum_i a_i = \sum_j b_j = G \quad \sum_{i=1}^{p} \sum_{j=1}^{q} \bar{c}_{ij} g_{ij} = C_g.$$

Most már a megfigyelési időpontra készített gravitációs modell az *I-divergencia* minimalizálással, mint kalibrációs elvvel a következő:

$$\left.\begin{array}{l} \min I(g, h) \\ \text{feltéve, hogy} \\ h_{ij} = r_i \exp(\alpha \bar{c}_{ij}) s_j \\ r_i, s_j > 0 \quad \text{minden } i, j\text{-re} \\ \sum_{i=1}^{p} \sum_{j=1}^{q} h_{ij} = G. \end{array}\right\} \text{(A)}$$

Látni fogjuk, hogy az (A) feladatban a h_{ij} mátrix sor- és oszlopösszegeire vonatkozó

$$\begin{cases} \sum_{j=1}^{q} h_{ij} = a_i, \\ \sum_{i=1}^{p} h_{ij} = b_j \end{cases} \tag{3.4}$$

feltételeket nem kell előre kikötnünk, mivel azok az optimális megoldásban úgyis teljesülni fognak. Az itt szereplő célfüggvényt kissé átalakítva:

$$I(g, h) = \frac{1}{G} \sum_{i,j} g_{ij} \log \frac{g_{ij}}{h_{ij}} = \frac{1}{G} \sum_{i,j} g_{ij} \log \frac{g_{ij}}{r_i e^{\alpha \bar{c}_{ij}} s_j}.$$

A célfüggvényből a konstans szorzót elhagyva az (A) feladat a következő alakban írható:

$$\min (D =) \sum_{i,j} g_{ij} \log \frac{g_{ij}}{r_i e^{\alpha \bar{c}_{ij}} s_j} \tag{3.5}$$

feltéve, hogy

$$r_i, s_j > 0 \quad \text{minden } i,j\text{-re}$$

$$\sum_{i,j} r_i e^{\alpha \bar{c}_{ij}} s_j = G. \tag{3.6}$$

A (3.6)-ban szereplő egyenlőség helyett vehetjük a következőt:

$$\sum_{i,j} r_i e^{\alpha \bar{c}_{ij}} s_j \leqq G \tag{3.7}$$

mivel a célfüggvény minimalizálása biztosítja azt, hogy az optimumban (3.7) egyenlőséggel teljesüljön. Az exponenciális függvény monotonitása miatt D minimalizálása helyettesíthető $\exp(D)$ minimalizálásával. Így mivel (3.5) alapján

$$\exp(D) = \prod_{i,j} g_{ij}^{g_{ij}} + \prod_{i,j} r_i^{-a_i} s_j^{-b_j} (e^{\alpha})^{-\bar{c}_{ij} g_{ij}}$$

innen a konstans első tagot elhagyva (A)-ból a következő feladatot nyerjük:

$$\left.\begin{array}{l} \min \prod_{i,j} r_i^{-a_i} s_j^{-b_j} (e^{\alpha})^{-\bar{c}_{ij} g_{ij}} \\ \text{feltéve, hogy} \\ r_i, s_j > 0 \quad \text{minden } i,j\text{-re} \\ (e^{\alpha} > 0 \text{ biztos teljesül}) \\ \sum_{i,j} \frac{1}{G} r_i s_j (e^{\alpha})^{\bar{c}_{ij}} \leqq 1 \end{array}\right\} \text{(A')}$$

(A')-ről már leolvasható, hogy egy (P)-nek megfelelő alakú geometriai programozási primál feladat (az r_i, s_j, e^{α} változókkal). A hozzátartozó (D)-nek megfelelő duál feladat — $\log v(\mathbf{y})$-t használva célfüggvényként — a következő lesz:

$$\max z(0 - \log z) + \sum_{i,j} y_{ij} \left(\log \frac{1}{G} + \log \frac{\sum_{k,s} y_{ks}}{y_{ij}} \right),$$

feltéve, hogy

$$y_{ij} \geqq 0 \quad \text{minden } i,j\text{-re}$$

$$z = 1$$

$$z a_i = \sum_j y_{ij}, \quad \text{minden } i\text{-re}$$

$$z b_j = \sum_i y_{ij}, \quad \text{minden } j\text{-re}$$

$$z\left(\sum_{i,j} \bar{c}_{ij} g_{ij}\right) = \sum_{i,j} \bar{c}_{ij} y_{ij}.$$

Egyszerű átalakítás, s a célfüggvényből konstans tagok elhagyása után a következő

alakba írható ez a feladat:

$$\left.\begin{array}{l} \max -\sum\limits_{i,j} y_{ij} \log y_{ij} \\ \text{feltéve, hogy} \\ y_{ij} \geqq 0, \quad \text{minden } i,j\text{-re} \\ \sum\limits_{j} y_{ij} = a_i, \quad \text{minden } i\text{-re} \\ \sum\limits_{i} y_{ij} = b_j, \quad \text{minden } j\text{-re} \\ \sum\limits_{i,j} \bar{c}_{ij} y_{ij} = C_g \end{array}\right\} \text{(B)}$$

Látható, hogy a (B) feladatnak van $\mathbf{y}>\mathbf{0}$ megoldása — pl. $y_{ij}=g_{ij}$ megfelel, hiszen a megfigyelt mátrix elemeiről feltételezhető a pozitivitás —, továbbá a célfüggvény felülről korlátos $\left(\text{hiszen } -y \log y \leqq \frac{1}{e}\right)$. Így a 2. („erős") dualitási tétel értelmében az (A′) célfüggvénye felveszi minimumát a megengedett tartományon. Az is világos, hogy az (A′) feladat a definiált értelemben szuperkonzisztens, hiszen a szereplő egyenlőtlenség-feltétel elég kis abszolút értékű r_i, s_j változók esetén szigorúan teljesül. Ezért, felhasználva az 1. („gyenge") dualitási tételt is, a (B) feladatnak van optimális megoldása, s egy optimális megoldás-párra fennállnak az egyensúlyi feltételek, melyek jelen esetben a következők:

$$\bar{y}_{ij} = \bar{r}_i \bar{s}_j e^{\bar{\alpha}\bar{c}_{ij}}, \quad \text{minden } i,j\text{-re.}$$

Ezt összevetve a (B) feladattal, azt kapjuk, hogy az (A) és (B) egy optimális megoldás párja kielégíti az alábbi feltételeket (aminek a megfordítottja is áll: az ilyen feltételeket kielégítő y_{ij}, ill r_i, s_j, α optimális megoldásai az (A), ill. (B)-nek)

$$\left.\begin{array}{l} y_{ij} = r_i s_j e^{\alpha \bar{c}_{ij}} \\ r_i, s_j > 0 \\ \sum\limits_{j} y_{ij} = a_i, \quad \text{minden } i\text{-re} \\ \sum\limits_{i} y_{.j} = b_j, \quad \text{minden } j\text{-re} \\ \sum\limits_{ij} \bar{c}_{ij} y_{ij} = C_g \end{array}\right\} \text{(C)}$$

Ez azt jelenti, hogy az *I-divergencia* minimalizálás — (A) feladat —, mint kalibrálási elv, ekvivalens az összköltség feltétellel — (C) feladat — az exponenciális ellenállásfüggvényű gravitációs modellnél, s mindkettő ekvivalens az entropiamaximalizálás összköltség feltétel melletti elvével — (B) feladat. S megkaptuk azt is, hogy a h_{ij} mátrixra vonatkozó (3.4) peremfeltételek valóban teljesülnek az optimális megoldásban.

3.3. *Az ellenállásfüggvény paraméterének numerikus meghatározása*

A kétszeresen korlátozott összköltség feltételes gravitációs modell ellenállásfüggvényében a paramétert — vagy paramétereket — hagyományosan próbálgatás jellegű heurisztikus módszerrel határozták meg. Ez már közepes ($p, q \approx 50$), de különösen nagy méretek ($p, q > 100$) esetén igen számításigényes, hiszen minden rögzített α-ra végre kell hajtani egy *Furness-típusú algoritmust*. Ezért tarthat érdeklődésre számot az a hatékony kalibrálási algoritmus, ami az exponenciális ellenállásfüggvény esetére alkalmazható, melyet az alábbiakban vázlatosan ismertetünk. További részletek ezzel kapcsolatban a [14] munkában találhatók.

Adott α esetén a (G) gravitációs probléma megoldása h_{ij}-re egyértelmű, s így definiálható a $h_{ij}(\alpha)$ függvény, s ezen keresztül a $C(\alpha)=\sum_{i,j} c_{ij}h_{ij}(\alpha)$ függvény.

A (G) feladathoz hozzárendelhetjük a

$$\min \sum_{i,j} c_{ij}h_{ij}, \quad \text{illetve} \quad \max \sum_{i,j} c_{ij}h_{ij}$$

feltéve, hogy $h_{ij} \geqq 0$ és

$$\sum_j h_{ij} = \bar{o}_i, \quad \text{minden } i\text{-re}$$

$$\sum_i h_{ij} = \bar{d}_j, \quad \text{minden } j\text{-re}$$

szállítási feladatpárt (h_{ij}-k a változók). Jelöljük a feladat optimális célfüggvényértékét $\underline{C}$-vel, ill. $\bar{C}$-vel. Nyilvánvaló, hogy bármely α-ra teljesül:

$$\underline{C} \leqq C(\alpha) \leqq \bar{C}.$$

EVANS bebizonyította — ld. [10]-ben —, hogy $C(\alpha)$ szigorúan monoton növekedő függvény, és

$$\lim_{\alpha \to -\infty} C(\alpha) = \underline{C}, \quad \lim_{\alpha \to \infty} C(\alpha) = \bar{C}.$$

Ezt a geometriai programozás dualitási tételével a [10]-ben szereplőnél egyszerűbb módon is be lehet látni — 1. [14]-ben. A gravitációs modell $h_{ij}=r_i e^{\alpha c_{ij}} s_j$ megoldásában az r_i, s_j értékek is egyértelművé tehetők, ha közülük egyet rögzítünk (ezért legyen pl. $r_1=1$).

Bevezetve az $e^{u_i}=r_i$, $e^{v_j}=s_j$ jelölést (így nyilván az u_i és v_j értékek egyaránt α függvényei):

$$u_1 = 0, \quad u_i = u_i(\alpha) \quad (i \neq 1), \quad v_j = v_j(\alpha).$$

Valamint:

$$h_{ij}(\alpha) = \exp\left(u_i(\alpha)+v_j(\alpha)+\alpha c_{ij}\right). \tag{3.8}$$

Belátható, hogy az itt szereplő függvények α szerint differenciálhatók, s a (3.8) egyenlőség differenciálásával — az α szerinti deriváltat vesszővel jelölve — ezt kapjuk:

$$\frac{dh_{ij}(\alpha)}{d\alpha} = h'_{ij}(\alpha) = [u'_i(\alpha)+v'_j(\alpha)+c_{ij}]h_{ij}(\alpha) \tag{3.9}$$

és nyilván

$$C'(\alpha) = \sum_{i,j} c_{ij} h'_{ij}(\alpha).$$

A (G) feladatban szereplő korlátozó feltételek α szerinti deriválásával kapjuk (az α argumentumot a rövidség kedvéért nem írjuk ki)

$$\sum_j h'_{ij} = 0, \quad i = 1, \ldots, p,$$

$$\sum_i h'_{ij} = 0, \quad j = 1, \ldots, q.$$

Ide behelyettesítve a (3.9) összefüggést, kapjuk:

$$\begin{cases} u'_i\left(\sum_j h_{ij}\right) + \sum_j v'_j h_{ij} + \sum_j c_{ij} h_{ij} = 0, & i = 1, \ldots, p, \\ v'_j\left(\sum_i h_{ij}\right) + \sum_i u'_i h_{ij} + \sum_i c_{ij} h_{ij} = 0, & j = 1, \ldots, q. \end{cases} \tag{3.10}$$

Könnyen látható, hogy $\alpha=0$ esetén a gravitációs modell megoldása a következő egyszerű alakban adódik:

$$h_{ij}(0) = \frac{\bar{o}_i \bar{d}_j}{T}, \tag{3.11}$$

ahol

$$T = \sum_i \bar{o}_i = \sum_j \bar{d}_j.$$

Így (3.10)-ből $\alpha=0$ esetére adódik:

$$u'_i \bar{o}_i + \sum_j v'_j \bar{o}_i \bar{d}_j / T + \sum_j c_{ij} \bar{o}_i \bar{d}_j / T = 0, \quad i = 1, \ldots, p,$$

$$v'_j \bar{d}_j + \sum_i u'_i \bar{o}_i \bar{d}_j / T + \sum_i c_{ij} \bar{o}_i \bar{d}_j / T = 0, \quad j = 1, \ldots, q.$$

Ezekből az egyenletekből, bevezetve a

$$K = \sum_{i,j} \bar{o}_i c_{ij} \bar{d}_j, \quad A_i = \sum_j c_{ij} \bar{d}_j, \quad B_j = \sum_i c_{ij} \bar{o}_i$$

jelöléseket a következőt kapjuk:

$$u'_i(0) + v'_j(0) = \frac{K}{T^2} - \frac{A_i + B_j}{T}, \quad \text{minden } i, j\text{-re.} \tag{3.12}$$

Behelyettesítve (3.12)-t (3.9)-be:

$$h'_{ij}(0) = \left(\frac{K}{T^2} - \frac{A_i + B_j}{T} + c_{ij}\right) \frac{\bar{o}_i \bar{d}_j}{T}, \quad \text{minden } i, j\text{-re.}$$

Erre az eredményre támaszkodva már könnyű kiszámítani $C'(0)$-t:

$$C'(0) = \sum_{i,j} \frac{\bar{o}_i c_{ij} \bar{d}_j}{T} \left(\frac{K}{T^2} - \frac{A_i + B_j}{T} + c_{ij}\right). \tag{3.13}$$

A $C(\alpha)$ függvényt az *elsőrendű Taylor sorával* közelítve:

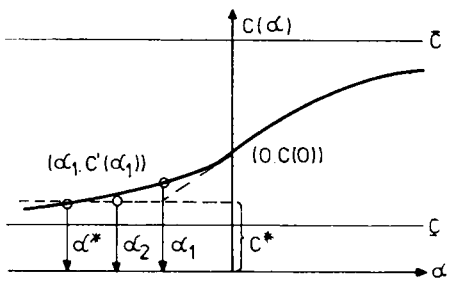
$$C(\alpha) \approx C(0)+\alpha C'(0). \tag{3.14}$$

Az összköltség-feltételre támaszkodó kalibrálásnál a feladatunk olyan α^*-ot keresni, melyre

$$C(\alpha^*) = C^*, \tag{3.15}$$

ahol C^* az előírt összköltség-szint. (3.14) és (3.15) felhasználásával α^* elsőrendű közelítésére ezt kapjuk:

$$\alpha^* \approx \alpha_1 = \frac{C^* - C(0)}{C'(0)}. \tag{3.16}$$



1. ábra

Az α_1 közelítő értéket úgy kaptuk, hogy a $C(\alpha)$ görbét az $\alpha=0$ helyen az érintőjével helyettesítettük — s ennek a metszéspontja a $C=C^*$ egyenessel adta α_1-et. Most a második közelítést, α_2-t a $(0, C(0))$ és $(\alpha_1, C(\alpha_1))$ pontokon átfektetett húr és a $C=C^*$ egyenes metszéspontjaként állíthatjuk elő (az algoritmust az 1. ábrán szemléltettük).

Egy α_0 közelítő érték pontossága $C(\alpha_0)$ C^*-tól vett relatív eltérése abszolút értékével mérhető:

$$E(\alpha_0) = \frac{|C(\alpha_0) - C^*|}{C^*}.$$

A megkövetelt pontossági kritérium:

$$E(\alpha_0) < \varepsilon,$$

ahol pl. $\varepsilon=0{,}01$ egy ésszerű hibahatár.

Sok feladatra (különösen a kis-közepes méretűekre, melyeknél $p, q<50$) az α_2 becslés már kielégítő pontosságot ad. Ha nem, akkor további „húr- és érintő-iterációkat" hajthatunk végre. Az utóbbiakat az teszi lehetővé, hogy $C'(\alpha)$, sőt az n-edik derivált $C^{(n)}(\alpha)$ is kiszámítható viszonylag kevés pótlólagos számítási ráfordítással tetszőleges α helyen — ennek bizonyítását l. a [14] cikkben.

Az α^* meghatározására szolgáló, ismertetett algoritmus számítógépes megvalósítása is megtörtént, FORTRAN nyelven készült programmal. Apróbb teszt-feladatok mellett megoldottunk a programmal próbafeladatként két, valós adatokra támaszkodó problémát. S ezzel készült *Budapest* forgalomelőrebecslési modell-

rendszerében a kerületi szintű ($p, q=22$) és az alkerületi szintű ($p, q=86$) forgalomszétosztási számításokban a paraméterbecslés.

Az első próbafeladat egy *Hollandia* 43 regiója közötti áruáramlatokat tartalmazó 43×43 méretű mátrixra vonatkozott. Itt az $\varepsilon=0{,}008$ relatív hibaszint eléréséhez csak az α_1 és α_2 elkészítésére volt szükség, s a hibaszint további csökkentése $\varepsilon=0{,}001$-re csak 3 további iterációt igényelt α-ra.

A második kísérleti számításnál *Magyarország* 123 régiója közötti közúti forgalmi mátrix volt a kiindulópont. Itt már megmutatkozott annak a jelentősége is, hogy $C'(\alpha)$ kiszámítására tetszőleges α esetén lehetőség van, ugyanis az $\varepsilon=0{,}005$ C-beli hibaszint eléréséhez az ε_2 meghatározása után még három további „érintő-iterációra" volt szükség, s ugyanakkor az ezek helyett kísérletképpen végrehajtott 5 „húr-iteráció" nem biztosította a megkívánt pontosságot, de több számítógép-időt igényelt.

Idáig még nem beszéltünk arról, hogy a kalibrációhoz szükséges, a jövőbeli állapotra vonatkozó C^* összköltség-szintet hogyan lehet előrebecsülni. Ennek megvalósítására többféle módszer is szóbajön. Itt ezek közül két, ésszerűnek látszó eljárást vázolunk. Az első esetben végrehajtjuk a forgalomelőrebecslést a jólismert multiplikatív növekedési tényezős modellel:

$$h_{ij} = r_i g_{ij} s_j, \quad \text{minden } i,j\text{-re}$$

$$\sum_j h_{ij} = \bar{o}_i, \quad \text{minden } i\text{-re}$$

$$\sum_i h_{ij} = \bar{d}_j, \quad \text{minden } j\text{-re.}$$

Majd a kapott $[h_{ij}]$ mátrixhoz az előrebecsült c_{ij} fajlagos utazási költségek segítségével kiszámítjuk a hozzátartozó $C=\sum_{i,j} c_{ij}h_{ij}$ utazási összköltséget, s ez lesz a C^*.

A másik eljárás a ténymátrixhoz tartozó $C_g=\sum_{i,j} \bar{c}_{ij}g_{ij}$ összköltségnek a

$$\min \sum_{i,j} \bar{c}_{ij}x_{ij}, \quad \text{illetve} \quad \max \sum_{i,j} \bar{c}_{ij}x_{ij}$$

$$\sum_j x_{ij} = o_i, \quad \text{minden } i\text{-re}$$

$$\sum_i x_{ij} = d_j, \quad \text{minden } j\text{-re}$$

szállítási feladatpár C_g, $\bar{C}_g$ optimális értékpárja közötti elhelyezkedését — ti. azt az arányt, ahogy a C_g a $(\underline{C}_g, \bar{C}_g)$ intervallumot felosztja — veszi a jövőre öröklődőnek, s C^*-ot úgy határozza meg, hogy ugyanilyen arányba ossza azt a $(C, \bar{C})$ intervallumot, amit az előrebecsült állapothoz tartozó szállítási feladat-pár:

$$\min \sum_{i,j} c_{ij}x_{ij}, \quad \text{illetve} \quad \max \sum_{i,j} c_{ij}x_{ij}$$

$$\sum_j x_{ij} = \bar{o}_i, \quad \text{minden } i\text{-re}$$

$$\sum_i x_{ij} = \bar{d}_j, \quad \text{minden } j\text{-re}$$

$(\underline{C}, \bar{C})$ optimális értékpárja ad.

1. Melléklet

Tömegközlekedési honnan hová mátrix *Budapest* 7 körzete között)
(lakás, munkahely)

Körzet	1	2	3	4	5	6	7
1	93	26	4	6	12	9	69
2	30	98	8	3	12	11	96
3	10	25	84	1	6	8	77
4	9	4	1	17	7	3	17
5	13	11	3	5	42	16	67
6	8	8	4	3	14	54	51
7	32	38	13	6	22	23	146

Előrebecsült kiinduló forgalom:

1: 243, 2: 218, 3: 249, 4: 98, 5: 127, 6: 123, 7: 175.

Előrebecsült befutó forgalom:

1: 197, 2: 188, 3: 135, 4: 44, 5: 101, 6: 114, 7: 454.

2. Melléklet

Növekedési tényezős módszerek eredményei

A módszereknél használt jelölések:

S1 sorösszeg

S2 oszlopösszeg

S1H sorösszeg hányados $\bar{o}_i/o_i^{(k)}$

S2H oszlopösszeg hányados $\bar{d}_j/d_j^{(k)}$

S1A $\sum_i (\bar{o}_i - o_i^{(k)})^2$

S2A $\sum_i (\bar{d}_j - d_j^{(k)})^2$

$$D = \frac{\sum_{i,j} (g_{ij} - h_{ij}^{(k)})^2}{\sum_{i,j} g_{ij}}.$$

Globális módszer

Körzet	1	2	3	4	5	6	7	S1
1	86,5	24,2	3,7	5,6	11,7	8,4	64,2	204,2
2	27,9	1,0	7,4	2,8	11,2	10,2	89,3	239,6
3	9,3	23,4	78,2	0,9	5,6	7,4	71,7	196,1
4	8,4	3,7	0,9	15,8	6,5	2,8	15,8	53,8
5	12,1	10,2	2,8	4,7	39,1	14,9	62,4	146,0
6	7,4	7,4	3,7	2,8	13,0	50,3	47,5	132,3
7	29,8	35,4	12,1	5,6	20,5	21,4	135,9	261,2
S2	180,7	195,8	108,9	38,3	107,4	115,1	488,2	

	1	2	3	4	5	6	7
S1H:	1,19	0,91	1,27	1,82	0,87	0,93	0,67
S2H:	1,09	0,96	1,24	1,15	0,94	0,99	0,93

S1A = 117,8 S2A = 45,8

Átlagtényezős módszer

Körzet	1	2	3	4	5	6	7	S1
1	99,8	27,1	4,4	6,4	12,5	9,5	71,4	231,2
2	29,5	93,6	8,2	3,0	11,4	10,6	90,8	247.0
3	10,9	26,5	94,9	1,1	6,3	8,6	81,0	229,4
4	11,2	4,9	1,3	21,2	8,5	3,7	20,5	71,2
5	12,6	10,4	3,0	4,9	39,3	15,3	62,5	148,0
6	7,9	7,7	4,1	2,9	13,4	52,6	48,6	137,3
7	28,7	33,0	12,2	5,4	19,0	20,2	125,4	243,8
S2	200,8	203,2	128,1	44,9	110,4	120,5	500,2	

	1	2	3	4	5	6	7
S1H:	1,05	0,88	1,09	1,38	0,86	0,90	0,72
S2H:	0,98	0,93	1,05	0,98	0,91	0,95	0,91

S1A = 86,4 S2A = 50,62

D = 3,77

Fratar módszer

Körzet	1	2	3	4	5	6	7	S1
1	105,1	28,7	4,9	5,9	13,0	10,5	74,2	242,2
2	26,2	83,6	7,5	2,3	10,0	9,9	79,6	219,2
3	11,6	28,5	105,1	1,0	6,7	9,6	85,3	247,9
4	15,8	6,9	1,9	26,0	11,8	5,4	28,5	96,3
5	10,9	9,0	2,7	3,7	33,9	13,8	53,6	127,6
6	7,1	6,9	3,8	2,3	11,9	49,1	42,9	124,0
7	20,7	24,0	9,1	3,9	13,7	15,3	89,7	175,8
S2	197,4	187,7	135,0	44,6	101,0	113,5	453,7	

	1	2	3	4	5	6	7
S1H:	1,00	0,99	1,00	1,02	1,00	0,99	1,00
S2H:	1,00	1,00	1,00	0,99	1,00	1,00	1,00

S1A = 2,81 S2A = 0,94

D = 1,64

Detroiti módszer

Kör-zet	1	2	3	4	5	6	7	S1
1	105,4	28,7	5,0	6,0	12,9	10,4	74,2	242,5
2	25,9	82,7	7,6	2,3	9,9	9,7	79,0	217,1
3	11,8	28,7	108,6	1,0	6,7	9,6	86,3	252,8
4	16,0	7,0	2,0	26,7	11,9	5,5	28,8	97,9
5	10,8	8,9	2,7	3,7	33,4	13,6	53,0	126,2
6	7,0	6,8	3,8	2,3	11,7	48,1	42,3	122,0
7	21,6	23,8	9,2	3,4	13,5	15,1	89,4	174,5
S2	197,7	186,6	138,8	45,4	100,1	111,9	452,7	

	1	2	3	4	5	6	7
S1H:	1,00	1,00	0,98	1,00	1,01	1,01	1,00
S2H:	1,00	1,01	0,97	0,97	1,01	1,02	1,00

S1A = 4,18 S2A = 5,06

D = 2,06

Furness módszer

Kör-zet	1	2	3	4	5	6	7	S1
1	105,3	28,9	4,8	5,9	13,2	10,5	74,4	343,1
2	25,9	83,0	7,2	2,3	10,0	9,8	78,9	217,1
3	12.0	29,6	106,2	1,0	7,0	10,0	88,4	254,2
4	15,5	6,7	1,8	25,5	11,7	5,4	28,0	94,6
5	10,7	8,9	2,6	3,6	33,6	13,7	52,7	125,8
6	7,0	6,9	3,7	2,3	12,0	49,4	43,0	124,4
7	20,4	23,8	8,7	3,3	13,6	15,2	88,7	173,8
S2	197.0	188,0	135,0	44,0	101,0	114,0	454,0	

	1	2	3	4	5	6	7
S1H:	1,00	1,00	0,98	1,04	1,01	0,99	1,01
S2H:	1,00	1,00	1,00	1,00	1,00	1,00	1,00

S1A = 6,64 S2A = 0,00

D = 2.07

IRODALOM

[1] BAKÓ, A.: „Forgalom előrebecslés növekedési tényezős módszerei", *KTMF Tudományos Közlemények* **3** (1980) 3—9.
[2] BAKÓ, A. és MARTON, L., „Forgalom előrebecslési módszerek összehasonlító vizsgálata és paraméterbecslése számítógéppel", *Kutatási zárójelentés*, KTMF, 1976.
[3] BEVIS, H. W., "Forecasting zonal traffic volumens", *Traffic Quarterly* (1956) 207—222.
[4] Брегман, Л., «Доказательство сходимостя Г. Б. Шелейковский для задачи с транспортными ограничениями», *Журнал Вычислимельной Математики и Математической Физики* **1** (1967) 147—156.
[5] BROKKE, G. E. and MERTZ, W. L., "Evaluating trip forecasting methods with an electronic computer", *Travel Characteristics in Urban Areas*, B **203** HRB (1957) 52—75.
[6] D'ESOPO, D. A. and LEFKOWITZ, L., "An algorithm for computing interzonal transfer using the gravity model", *Operation Research* **11** (1963) 901—907.
[7] DUFFIN, R., PETERSON, E. and ZENER, C., *Geometric Programming: Theory and Application* (Wiley, New York, 1967).
[8] CSISZÁR, I., "I-divergence geometry of probability distributions and minimization problems", *Annales of Probability* **3** (1975) 146—158.
[9] EVANS, A. W., "Some properties of trip distribution methods", *Transportation Research* **4** (1970) 19—36.
[10] EVANS, S. P., "A relationship between the gravity model for trip distribution and the transportation problem in linear programing", *Transportation Research* **7** (1973) 39—61.
[11] FRATAR, T. J., "Vehicular trip distribution by successive approximation", *Traffic Quarterly* (1954) 53—65.
[12] HUTCHINSON, B. G., *Principle of Urban Transportation Systems Planning* (Scripta Book Company, Washington, 1978).
[13] KÁDAS, S., „A geometriai programozás egy megoldási módszere", *Alkalmazott Matematikai Lapok* **2** (1976) 67—81.
[14] KÁDAS, S. and KLAFSZKY, E., "Estimation of the parameters in the gravity model for trip distribution: A new method and solution algorithm", *Regional Science and Urban Economics* **6** (1976) 439—457.
[15] KIRBY, H. R., "Normalising factors of the gravity model-an interpretation", *Transportation Research* **4** (1970) 37—50.
[16] KIRBY, H. R. and LEESE, M. N., "Trip distribution calculation and sampling errors: Some theoretical aspects", *Environment and Planning* **10** (1968) 837—851.
[17] KLAFSZKY, E., „Az Input-Output tábla előrebecsléséről", *MTA SZTAKI Közlemények* **10** (1973) 1—13.
[18] KOLLER, S., *Forgalomtechnika* (Tankönyvkiadó, Budapest, 1967).
[19] MONIGL, J., *Városi forgalomszétosztási modellek kialakítása* (KÖTUKI, 1976).
[20] MORRIS, R. J., *A Comparative Analysis of Trip Distribution and Traffic Assignment Models for Tansportation Planning in Developing Regions* (Stanford University, 1973).
[21] STOPHER, R. R., *Urban Transportation Modeling and Planning* (Lexigton Books, Lexigton, Massachusetts, 1975).
[22] SNICKARS, F. and WEIBULL, J. W., "A minimum information principle", *Regional Science and Urban Economics* **7** (1977) 137—168.
[23] WILSON, A. G., *Entropy in Urban and Regional Modelling* (Pion Limited, 1970).

(Beérkezett: 1981. február 16.)

BAKÓ ANDRÁS
KÖZLEKEDÉSI ÉS TÁVKÖZLÉSI MŰSZAKI FŐISKOLA
9020 GYŐR, SÁGVÁRI ENDRE U.

KÁDAS SÁNDOR
MATX KÁROLY KÖZGAZDASÁGTUDOMÁYI EGYETEM
1093 BUDAPEST, DIMITROV TÉR 8.

OPTIMIZATION PROBLEMS OF TRAFFIC DISTRIBUTION

A. BAKÓ and S. KÁDAS

The main components of the traffic planning problem are traffic forecasting and distribution. A lot of algorithms are known for solving these problems.

We summarize the two main methods. One of them is the growth factor method, the other is the gravity type method. A new calibration technique of the gravity model is also presented. In connection with the graity model we prove that the I-divergence minimization problem rest the entropy maximization problem lead to the same results. At the end of the paper some of ouí computational results are summarized.

Alkalmazott Matematikai Lapok 7 (1981) 139—157

A LINEÁRIS CSEREMODELL EGYENSÚLYI ÁRÁNAK MEGHATÁROZÁSA GEOMETRIAI PROGRAMOZÁSSAL

KLAFSZKY EMIL

Miskolc

A dolgozatban a lineáris cseremodell (GALE [2]) egyensúlyi pontját vizsgáljuk. A dolgozat öt részre tagozódik:

Az első részben a modell leírását adjuk. Majd — a lineáris programozás dualitási tételét használva — megmutatjuk, hogy a cseremodell egyensúlyi pontjának problémája ekvivalens egy komplementaritási feladattal.

A második részben EISENBERG—GALE [1] azon eredményét, amely szerint az egyensúlyi pont karakterizálható egy matematikai programozási feladattal (nevezzük ezt a matematikai programozási feladatot *Gale feladatnak*), egy gyengébb formában (azzal a feltevéssel, hogy létezik egyensúlyi pont), egyszerű módon — a geometriai egyenlőtlenség felhasználásával — igazoljuk.

A harmadik részben két olyan matematikai programozási feladatot adunk ((I) és (II) feladat) amelyekről — a geometriai programozás dualitási tételét használva [3] — megmutatjuk, hogy különkülön karakterizálják az egyensúlyi pontot. Ezen eredményből egyszerűen adódik EISENBERG—GALE eredménye is.

A negyedik részben kísérletet teszünk mindhárom optimalizációs feladat (*Gale feladat*, (I) és (II) feladat) gazdasági interpretálására, majd megmutatjuk a *Gale feladat* és a (II) feladat dualitási kapcsolatát.

Az ötödik részben a (II) feladat segítségével és a *Kőnig—Hall modell* felhasználásával [3] egy egyszerű — a szállítási feladat magyar módszeréhez hasonló — eljárást adunk a lineáris cseremodell egyensúlyi pontjának meghatározására.

1. A modell leírása

Tekintsünk egy gazdaságot, ahol egy *termelő* — n különböző — $G_1, G_2, \ldots, G_n$ *árut* termel. Egy termelési periódus alatt a G_j áruból γ_j *egységnyi mennyiséget* állít elő.

A gazdaságban van m számú önálló *fogyasztó*, jelöljük ezeket a $C_1, C_2, \ldots, C_m$ szimbólumokkal. A C_i fogyasztónak a termelési periódus alatt β_i *jövedelem* áll rendelkezésére, hogy áruk vásárlására költse. Tegyük fel, hogy ismerjük a nemnegatív α_{ij} számokat *(utilitási mátrix)*, amelyek azt mutatják, hogy a C_i *fogyasztó számára mennyit ér a* G_j *áru egy egysége.*

Az $\mathbf{A}=(\alpha_{ij})$ $m\times n$-es hasznossági mátrix, a $\mathbf{b}=(\beta_i)$ m-es jövedelem vektor és a $\mathbf{c}=(\gamma_j)$ n-es áru vektor a cseremodell paraméterei, amelyekre az alábbi természetes megkötéseket tesszük:

Az $\alpha_{ij}\geqq 0$ nem negatív és

(i) minden oszlopban van legalább egy pozitív elem, azaz $\sum_{i=1}^{m}\alpha_{ij}>0$, $\forall j$-re.

(ii) minden sorban van legalább egy pozitív elem, azaz $\sum_{j=1}^{m}\alpha_{ij}>0$, $\forall i$-re.

A $\beta_i>0$ minden i-re és $\gamma_j>0$ minden j-re.

Jelölje $\mathbf{p}=(\pi_1, \ldots, \pi_j, \ldots, \pi_n)\geqq 0$ nem-negatív koordinátájú vektor a $G_1, \ldots, G_j, \ldots, G_n$ árufajták egység árait. És jelölje az $\mathbf{X}=(\xi_{ij})\geqq 0$ $m\times n$-es nem-negatív elemű mátrix a vásárolt mennyiségeket. Azaz ξ_{ij} az a mennyiség amennyit a C_i fogyasztó a G_j áruból vásárol. Ezen változókra az alábbi csere egyensúly megkötéseknek kell teljesülni:

Pénz egyensúly:

$$\sum_{j=1}^{n} \pi_j \xi_{ij} = \beta_i, \quad (i = 1, \ldots, m), \tag{1.1}$$

azaz a C_i fogyasztó által elköltött pénz egyezzék meg az ő pénz készletével.

Áru egyensúly:

$$\sum_{i=1}^{m} \xi_{ij} = \gamma_j, \quad (j = 1, \ldots, n), \tag{1.2}$$

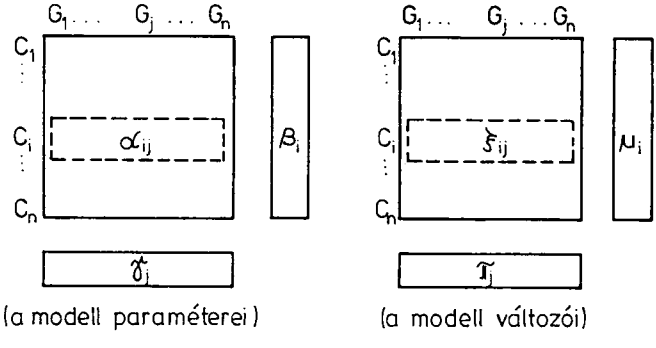
azaz a G_j áruból megvásárolt összes mennyiség egyezzék meg a kínálattal.

A kétféle egyensúlyi feltételből egyszerű szummázással kapjuk:

Össz kereslet-kínálat egyensúly:

$$\sum_{j=1}^{n} \gamma_j \pi_j = \sum_{i=1}^{m} \beta_i. \tag{1.3}$$

A cseremodell paramétereit és változóit az alábbi sémán szemléltetjük: (Az $\mathbf{u}=(\mu_1, \ldots, \mu_i, \ldots, \mu_m)$ változót később ismertetjük.)



1. ábra

A leírás egyszerűsítése érdekében az **A** mátrix *i*-edik sorát $\mathbf{a}_i$ vektorral; hasonlóan az **X** mátrix *i*-edik sorát $\mathbf{x}_i$ vektorral jelöljük. (Vektorok jelölésére latin betűt, skalár jelölésére görög betűt használunk.)

Azzal a linearitási feltevéssel élünk, hogy ha a C_i fogyasztó vásárlási vektora $\mathbf{x}_i$, akkor a vásárló $\mathbf{a}_i\mathbf{x}_i$ haszonhoz jut. Ez az — eléggé erőszakosnak tűnő — feltételezés indokolja a *lineáris* cseremodell elnevezést.

Nézzük most, hogy hogyan viselkedik a C_i fogyasztó: Adott $\mathbf{p}$ árvektor esetén a C_i fogyasztó számára az $\mathbf{x}_i^* \geqq \mathbf{0}$ vásárlási vektor *elfogadható* ha az összes $\mathbf{x}_i \geqq \mathbf{0}$, $\mathbf{p}\mathbf{x}_i = \beta_i$ ((1.1) feltétel) feltételt kielégítő $\mathbf{x}_i$ vektorok között az $\mathbf{a}_i\mathbf{x}_i$ értéket (hasznot) maximalizálja.

DEFINÍCIÓ. A $\mathbf{p} \geqq \mathbf{0}$ árvektort *egyensúlyi árnak* nevezzük, ha létezik olyan $\mathbf{X} \geqq \mathbf{0}$ vásárlási mátrix, amely az (1.1), (1.2) egyensúlyi megkötéseket teljesíti és minden C_i fogyasztó részére elfogadható. (A $\mathbf{p}$ egyensúlyi árhoz adott $\mathbf{X}$ elfogadható mátrixot egyensúlyi vásárlási mátrixnak, vagy röviden *egyensúlyi vásárlásnak* nevezzük).

Célszerű kihangsúlyozni, hogy ha $\mathbf{p}$ egyensúlyi ár, akkor nem kívánjuk meg, hogy minden az $\mathbf{a}_i\mathbf{x}_i$ értéket maximalizáló $\mathbf{x}_i$ ($\mathbf{x}_i \geqq \mathbf{0}$, $\mathbf{p}\mathbf{x}_i = \beta_i$ feltétel esetén) egyensúlyi vásárlás is legyen, hanem csak azt, hogy az egyensúlyi vektor egy maximalizáló vektor legyen.

A definícióból látható, hogy az egyensúlyi $\mathbf{p}$ árvektorra és az egyensúlyi $\mathbf{X}$ vásárlási mátrixra nincs hatással az, hogy ha az $\mathbf{A} = (\alpha_{ij})$ utilitási mátrix bármelyik sora helyett annak valamely pozitív számmal vett szorzatát tekintjük. Ez mutatja, hogy modellünkben az α_{ij} nem tényleges értékelőt, hanem csak értékelési arányokat fejez ki. A modellel kapcsolatban a következő kérdés vetődik fel: Létezik-e egyensúlyi árvektor és az egyértelmű-e (egzisztencia, unicitás); valamint, ha létezik, hogyan lehet azt meghatározni (algoritmus)?

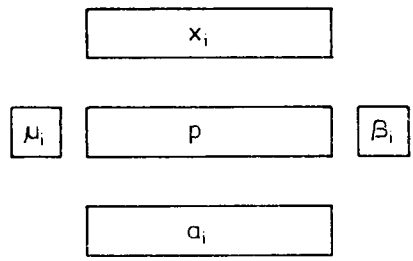
Mielőtt a felvetett kérdésekre a következő fejezetekben válaszolnánk előbb megmutatjuk, hogy a lineáris programozás dualitási tételét használva egy komplementaritási jellegű szükséges és elégséges feltételt nyerhetünk az egyensúlyi árra és az egyensúlyi vásárlásra (az úgynevezett egyensúlyi pontra).

1.1. LEMMA. A $\mathbf{p} = (\pi_j)$ vektor és az $\mathbf{X} = (\xi_{ij})$ mátrix akkor és csak akkor egyensúlyi, ha létezik $\mathbf{u} = (\mu_1, \ldots, \mu_i, \ldots, \mu_m)$ vektor úgy, hogy a:

$$
(1.4)\qquad \begin{cases} \xi_{ij} \geqq 0, \quad \pi_j > 0, \quad \mu_i > 0, \quad \text{minden } i,j\text{-re}, \\ \sum_{j=1}^{n} \pi_j \xi_{ij} = \beta_i, \quad \text{minden } i\text{-re}, \\ \mu_i \pi_j \geqq \alpha_{ij}, \quad \text{minden } i,j\text{-re}, \\ (\mu_i \pi_j - \alpha_{ij})\xi_{ij} = 0, \quad \text{minden } i,j\text{-re}, \\ \sum_{i=1}^{m} \xi_{ij} = \gamma_j, \quad \text{minden } j\text{-re} \end{cases}
$$

feltételeket kielégíti.

Bizonyítás: A $\mathbf{p}$ árvektor ismeretében a C_i fogyasztó stratégiája egy egyszerű lineáris programozási feladat:



2. ábra

Keresendő azon $\mathbf{x}_i$ vektor, amely a

$$\mathbf{p}\mathbf{x}_i = \beta_i,$$

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

feltételek esetén az

$$\mathbf{a}_i\mathbf{x}_i$$

célfüggvényt maximalizálja.

Ennek duálja: Keresendő azon μ_i szám, amely a

$$\mu_i\pi_j \geqq \alpha_{ij}, \quad (j = 1, \ldots, n)$$

feltételek esetén a

$$\mu_i\beta_i$$

értéket minimalizálja.

A dualitási tétel szerint $\mathbf{x}_i$, μ_i akkor és csak akkor optimális megoldás pár, ha a primál feltételt is, a duál feltételt is kielégíti és még az egyensúlyi

$$(\mu_i\pi_j - \alpha_{ij})\xi_{ij} = 0, \quad (j = 1, \ldots, n)$$

feltételeket is teljesíti.

A primál-, a duál- és az egyensúlyi feltételnek minden i-re teljesülni kell.

Minthogy $\alpha_{ij} \geqq 0$ és $\pi_j \geqq 0$ így az (i) és (ii) feltételek miatt $\pi_j > 0$ és $\mu_i > 0$. ∎

Megjegyzés: A $(\mu_i\pi_j - \alpha_{ij})\xi_{ij} = 0$ komplementaritási feltételből kiindulva felhasználva (1.1)-et adódik, hogy:

$$\mu_i = \frac{\mathbf{a}_i\mathbf{x}_i}{\beta_i}. \tag{1.5}$$

A C_i fogyasztó $\mathbf{a}_i\mathbf{x}_i$ haszonhoz jut β_i érték ráfordítással; így μ_i *haszonrátának* tekinthető.

2. Az egyensúlyi pont meghatározása, mint matematikai programozási feladat

Az egyensúlyi vásárlási mátrix minden fogyasztó számára maximalizálja a hasznot. A számtani-mértani közép egyenlőtlenséget használva egy optimalizálási jellegű szükséges feltételt adunk az egyensúlyi vásárlásra; azaz megmutatjuk, hogy az egyensúlyi vásárlási mátrix egy egycélú optimalizálási feladatot is megold.

2.1. LEMMA. Ha *van egyensúlyi vásárlási mátrix*, akkor valamely **X** mátrix akkor és csak akkor egyensúlyi, ha a

$$\begin{cases} \sum_{i=1}^{m} \xi_{ij} = \gamma_j, & (j = 1, \ldots, n), \\ \xi_{ij} \geqq 0, & (i = 1, \ldots, m;\ j = 1, \ldots, n) \end{cases} \tag{2.1}$$

feltételek esetén maximalizálja a

$$\prod_{i=1}^{m} (\mathbf{a}_i \mathbf{x}_i)^{\beta_i} \tag{2.2}$$

függvényt.

Bizonyítás: Legyen $\mathbf{X}^*$ egyensúlyi vásárlási mátrix és $\mathbf{p}^*$ az egyensúlyi ár, valamint $\mathbf{u}^*$ az egyensúlyi haszonráta vektor.

Legyen **X** mátrix a (2.1) feltételt kielégítő mátrix. Minthogy $\mathbf{p}^*, \mathbf{u}^*$ egyensúlyi ezért $\alpha_{ij} \leqq \mu_i^* \pi_j^*$. Ezt felhasználva kapjuk, hogy

$$\prod_{i=1}^{m} \left(\sum_{j=1}^{n} \alpha_{ij} \xi_{ij} \right)^{\beta_i} \leqq \prod_{i=1}^{m} \left(\sum_{j=1}^{n} (\mu_i^* \pi_j^*) \xi_{ij} \right)^{\beta_i}, \tag{2.3}$$

és egyenlőség akkor és csak akkor, ha

$$(\mu_i^* \pi_j^* - \alpha_{ij}) \xi_{ij} = 0, \quad (i = 1, \ldots, m;\ j = 1, \ldots, n). \tag{2.4}$$

A (2.3) jobb oldalába μ_i^* értékét (1.5) szerint beírva és átrendezve:

$$\prod_{i=1}^{m} \left(\sum_{j=1}^{n} (\mu_i^* \pi_j^*) \xi_{ij} \right)^{\beta_i} = \prod_{i=1}^{m} (\mathbf{a}_i \mathbf{x}_i^*)^{\beta_i} \cdot \left(\frac{\sum_{j=1}^{n} \pi_i^* \xi_{ij}}{\beta_i} \right)^{\beta_i}. \tag{2.5}$$

De a geometriai egyenlőtlenség miatt:

$$\prod_{i=1}^{m} \left(\frac{\sum_{j=1}^{n} \pi_j^* \xi_{ij}}{\beta_i} \right)^{\beta_i} \leqq \left(\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \pi_j^* \xi_{ij}}{\sum_{i=1}^{m} \beta_i} \right)^{\sum_{i=1}^{m} \beta_i} \tag{2.6}$$

és egyenlőség akkor és csak akkor, ha

$$\left(\sum_{j=1}^{n} \pi_j^* \xi_{ij} \right) \cdot \sum_{i=1}^{m} \beta_i = \beta_i \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \pi_j^* \xi_{ij} \right), \quad (i = 1, \ldots, m). \tag{2.7}$$

Minthogy ξ_{ij} a (2.1) feltételt kielégíti, így (1.3) miatt

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \pi_j^* \xi_{ij} = \sum_{i=1}^{m} \beta_i.$$

Ezt (2.6) és (2.7)-be helyettesítve az alábbi egyszerűbb formát kapjuk:

$$\prod_{i=1}^{m}\left(\frac{\sum_{j=1}^{n}\pi_j^*\xi_{ij}}{\beta_i}\right)^{\beta_i} \leqq 1, \tag{2.8}$$

és egyenlőség akkor és csak akkor, ha

$$\sum_{j=1}^{n}\pi_j^*\xi_{ij} = \beta_i, \quad (i = 1, \ldots, m). \tag{2.9}$$

A fentieket összefoglalva: a (2.3), (2.5) és (2.8)-ból kapjuk, hogy:

$$\prod_{i=1}^{m}(\mathbf{a}_i\mathbf{x}_i)^{\beta_i} \leqq \prod_{i=1}^{m}(\mathbf{a}_i\mathbf{x}_i^*)^{\beta_i} \tag{2.10}$$

és — (2.4), (2.9)-ből — egyenlőség akkor és csak akkor, ha

$$(\mu_i^*\pi_j^* - \alpha_{ij})\xi_{ij} = 0, \quad (i = 1, \ldots, m;\ j = 1, \ldots, n)$$

$$\sum_{j=1}^{n}\pi_j^*\xi_{ij} = \beta_i, \qquad (i = 1, \ldots, m). \tag{2.11}$$

A (2.10) adja, hogy ha egyensúlyi az $\mathbf{X}^*$ mátrix, akkor maximalizálja a (2.2) függvényt. A másik irány ha $\bar{\mathbf{X}}$ optimalizálja (2.2)-t, akkor (2.11 fenn áll, vagyis $\bar{\mathbf{X}}$, $\mathbf{p}^*$, $\mathbf{u}^*$ egyensúlyi pont az 1.1. lemmából következően.

A 2.1. lemma egy erős megkötés („ha egyáltalán van egyensúlyi pont") esetén ad szükséges és elégséges feltételt az egyensúlyi pontra. E. EISENBERG és D. GALE [1] ezen erős megkötés nélkül bizonyítja a lemmát és megmutatja, hogy az egyensúlyi ár egyértelmű (unicitás) és egy egyszerű formulát ad — az egyensúlyi vásárlás ismeretében — a meghatározására, azaz az alábbi tételt adják:

(i) Az $\mathbf{X}^*$ mátrix akkor és csak akkor egyensúlyi, ha (2.1) feltétel esetén a (2.2) függvényt maximalizálja.

(ii) Az $\mathbf{X}^*$ egyensúlyi vásárlás ismeretében, az egyensúlyi ár:

$$\pi_j^* = \max_{(i)}\frac{\alpha_{ij}}{\mu_i^*}, \quad (j = 1, \ldots, n) \tag{2.12}$$

$$\left(\text{ahol } \mu_i^* = \frac{\mathbf{a}_i\mathbf{x}_i^*}{\beta_i} \text{ az egyensúlyi ráta}\right).$$

(iii) A $\mathbf{p}^*$ egyensúlyi ár egyértelmű.

A következő fejezetben az egyensúlyi pontot karakterizáló tételünk következményeként adjuk majd EISENBERG—GALE fenti eredményét.

3. Az egyensúlyi pont karakterizálása geometriai programozással

A következőkben megmutatjuk, hogy az egyensúlyi pont karakterizálható úgy is mint egy primál és duál geometriai programozási feladatpár optimális megoldás párja.

(I) feladat: Meghatározandó azon $\mathbf{X} \geqq \mathbf{0}$ mátrix és $\mathbf{p} \geqq \mathbf{0}$ vektor, melyekre a

$$\sum_{j=1}^{n} \pi_j \xi_{ij} = \beta_i, \quad (i = 1, \ldots, m), \tag{3.1}$$

$$\sum_{i=1}^{m} \xi_{ij} = \gamma_j, \quad (j = 1, \ldots, n) \tag{3.2}$$

feltételek fennállnak és a

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \pi_j \xi_{ij} \log \frac{\pi_j}{\alpha_{ij}} \tag{3.3}$$

célfüggvényt minimalizálják.

(II) feladat: Meghatározandó azon $\mathbf{u} > \mathbf{0}$ és $\mathbf{v} > \mathbf{0}$ vektorok, melyekre az

$$\alpha_{ij} \leqq \mu_i \nu_j, \quad (i = 1, \ldots, m; j = 1, \ldots, n), \tag{3.4}$$

$$\sum_{j=1}^{n} \nu_j \gamma_j = \sum_{i=1}^{m} \beta_i, \quad (i = 1, \ldots, m) \tag{3.5}$$

feltételek fennállnak és a

$$\sum_{i=1}^{m} \beta_i \log \frac{1}{\mu_i} \tag{3.6}$$

célfüggvényt maximalizálják.

A két feladat közt fennáll az alábbi dualitási tétel:

3.1. TÉTEL. 1° Mind az (I) feladatnak, mind a (II) feladatnak van optimális megoldása.

2° A (3.1) és (3.2) primál feltételnek eleget tevő $\mathbf{X}^* \geqq \mathbf{0}$ mátrix és $\mathbf{p}^* \geqq \mathbf{0}$ vektor, valamint a (3.4) és (3.5) duál feltételnek eleget tevő $\mathbf{u}^* > \mathbf{0}$ és $\mathbf{v}^* > \mathbf{0}$ vektorok akkor és csak akkor optimalizálják a (3.3) illetve (3.6) célfüggvényt, ha a

$$\mathbf{p}^* = \mathbf{v}^*,$$

és a

$$(\mu_i^* \nu_j^* - \alpha_{ij}) \xi_{ij}^* = 0, \quad (i = 1, \ldots, m; \; j = 1, \ldots, n)$$

optimalitási feltételeket is teljesítik, azaz ha $\mathbf{p}^*$, $\mathbf{X}^*$, $\mathbf{u}^*$ a lineáris cseremodell egyensúlyi pontja.

Bizonyítás: Az (I) feladat paramétereit és változóit új paraméterekkel és változókkal helyettesítjük, majd a célfüggvényt átalakítjuk, hogy ily módon egy standard geometriai programozási duál feladatot nyerjünk.

Jelöljük:

$$\beta = \sum_{i=1}^{m} \beta_i.$$

Legyen:

$$(3.7) \qquad \begin{cases} \bar{\beta}_i = \dfrac{\beta_i}{\beta}, & (i = 1, \ldots, m), \\ \bar{\alpha}_{ij} = \dfrac{\alpha_{ij}\gamma_j}{\beta}, & (i = 1, \ldots, m;\ j = 1, \ldots, n), \end{cases}$$

$$(3.8) \qquad \begin{cases} \bar{\pi}_j = \dfrac{\pi_j \gamma_j}{\beta}, & (j = 1, \ldots, n), \\ \bar{\xi}_{ij} = \dfrac{\pi_j \xi_{ij}}{\beta}, & (i = 1, \ldots, m;\ j = 1, \ldots, n). \end{cases}$$

Így a (3.1) és (3.2) feltételek:

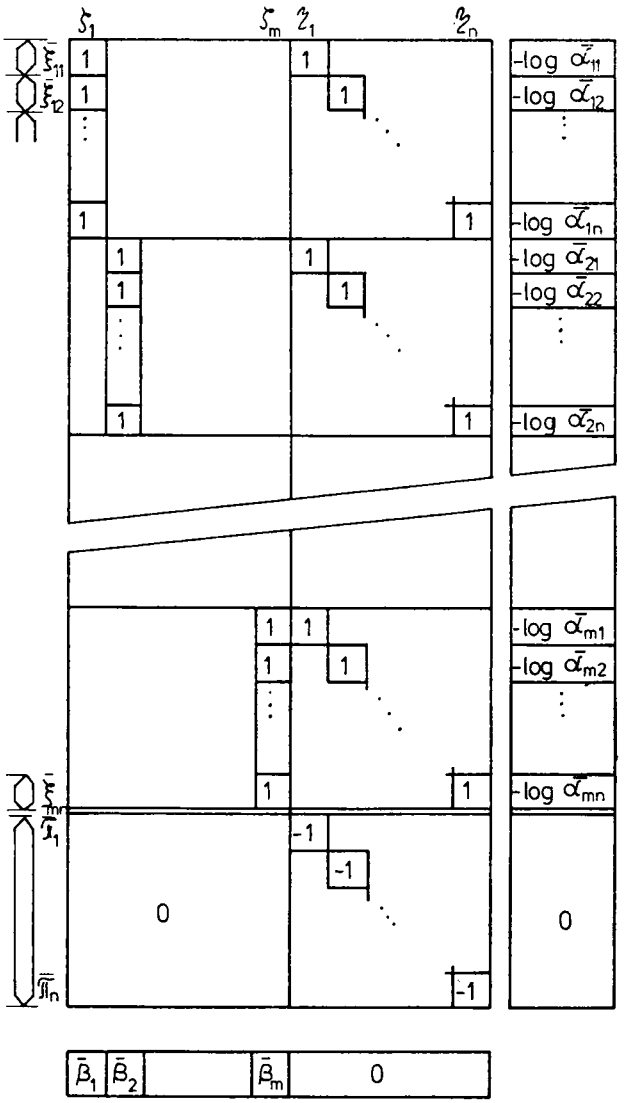
$$(3.9) \qquad \sum_{j=1}^{n} \bar{\xi}_{ij} = \bar{\beta}_i, \quad (i = 1, \ldots, m),$$

$$(3.10) \qquad \sum_{i=1}^{m} \bar{\xi}_{ij} = \bar{\pi}_j, \quad (j = 1, \ldots, n).$$

Ezen jelöléssel a $\sum_{j=1}^{n} \bar{\pi}_j = 1$ és a $\sum_{i=1}^{m} \bar{\beta}_i = 1$ normálás teljesül. A (3.3) célfüggvény pedig — egy β szorzótól eltekintve:

$$(3.11) \qquad \sum_{i=1}^{m} \sum_{j=1}^{n} \bar{\xi}_{ij}(-\log \bar{\alpha}_{ij}) + \log \frac{\prod_{j=1}^{n} \bar{\pi}_j^{\bar{\pi}_j}}{\left(\sum_{j=1}^{n} \bar{\pi}_j\right)^{\sum_{j=1}^{m} \bar{\pi}_j}} \to \min!$$

Most már világosan adódik, hogy ez egy standard geometriai programozási duál feladat. Abból a célból, hogy párját, a geometriai programozási primál feladatot könnyen felírhassuk, sematikusan adjuk együttható táblázatát:



3. ábra

Ennek párja egy standard primál geometriai programozási feladat: a $\zeta_1, \ldots, \zeta_m$; $\eta_1, \ldots, \eta_n$ változókra az

$$e^{\zeta_i+\eta_j+\log \bar{\alpha}_{ij}} \leqq 1, \quad (i = 1, \ldots, m; j = 1, \ldots, n), \tag{3.12}$$

$$\sum_{j=1}^{n} e^{-\eta_j} \leqq 1 \tag{3.13}$$

feltételek teljesüljenek és maximalizálják a

$$\sum_{i=1}^{m} \bar{\beta}_i \zeta_i \tag{3.14}$$

célfüggvényt.

Mindkét feladat a *Slater-féle regularitást* teljesíti, így mindegyiknek van optimális megoldása és ezekre a geometriai programozás dualitási tétele szerint fennáll az optimalitási feltétel [3]:

$$e^{\zeta_i+\eta_j+\log \bar{a}_{ij}} \bar{\xi}_{ij} = \bar{\xi}_{ij}, \quad (i = 1, \ldots, m; j = 1, \ldots, n), \tag{3.15}$$

$$e^{-\eta_j} \sum_{j=1}^{n} \bar{\pi}_j = \bar{\pi}_j, \quad (j = 1, \ldots, n). \tag{3.16}$$

Jelöljük: $e^{-\zeta_i}=\mu_i$ és $e^{-\eta_i}=\frac{\nu_j\gamma_j}{\beta}$, $(\mu_i>0, \nu_j>0)$.

Így a (3.16) és (3.15) optimalitási kritérium (felhasználva (3.7) és (3.8)-at, valamint azt, hogy $\sum_j \bar{\pi}_j=1$):

$$\pi_j = \nu_j, \quad (j = 1, \ldots, n), \tag{3.17}$$

$$(\mu_i \nu_j - \alpha_{ij})\xi_{ij} = 0, \quad (i = 1, \ldots, m; j = 1, \ldots, n). \tag{3.18}$$

A (3.18)-ban már (3.17)-et is felhasználtuk.

A (3.12) és (3.13) primál feltételek:

$$\alpha_{ij} \leqq \mu_i \nu_j, \quad (i = 1, \ldots m; j = 1, \ldots n), \tag{3.19}$$

$$\sum_{j=1}^{n} \nu_j \gamma_j \leqq \sum_{i=1}^{m} \beta_i.$$

De (3.16) miatt ez utóbbi az optimumnál egyenlőséggel teljesül, így elég csak abban az esetben tekinteni, azaz

$$\sum_{j=1}^{n} \nu_j \gamma_j = \sum_{i=1}^{m} \beta_i. \tag{3.20}$$

A (3.14) alatti primál célfüggvény:

$$\frac{1}{\beta} \sum_{i=1}^{m} \beta_i \log \frac{1}{\mu_i}. \tag{3.21}$$

A (3.19) és (3.20) adja a (3.4) és (3.5) feltételeket, (3.21) pedig a (3.6) célfüggvényt. A (3.17) és (3.18) optimalitási kritérium pedig a tétel állítását adja.

Minthogy $\mu_i=e^{-\zeta_i}$ és $e^{-\eta_j}=\frac{\nu_j\gamma_j}{\beta}$, így az optimális **u** és **v** pozitívak. ∎

KÖVETKEZMÉNY. Az egyensúlyi $\mathbf{p}^*$ ár és $\mathbf{u}^*$ ráta pozitív és egyértelmű, valamint

$$\mu_i^* = \frac{\mathbf{a}_i \mathbf{x}_i^*}{\beta_i}; \tag{3.22}$$

$$\pi_j^* = \max_{(i)} \frac{\alpha_{ij}}{\mu_i^*}; \quad (j = 1, \ldots, n). \tag{3.23}$$

Bizonyítás: A tételből azonnal adódik. Minthogy $\mathbf{p}^* = \mathbf{v}^*$ így $\mathbf{p}^*$ egyértelmű. A (3.22) a $(\mu_i^* v_j^* - \alpha_{ij})\xi_{ij}^* = 0$ optimalitásból adódik egyszerű összegzéssel, mint (1.5)-nél már láttuk, és ebből $\mathbf{u}^*$ egyértelműsége nyilvánvaló. Minthogy $\pi_j^* \mu_i^* \geqq \alpha_{ij}$, ezért $\pi_j^* \geqq \max_{(i)} \frac{\alpha_{ij}}{\mu_i^*}$. De itt egyenlőtlenség nem lehet a $\sum_{i=1}^{m} \beta_i \log \frac{1}{\mu_i^*}$ maximum tulajdonsága miatt. ∎

Megjegyzés. E. EISENBERG—D. GALE [1] eredménye is azonnal adódik, minthogy a tétel garantálja, hogy egyensúlyi pont mindig van.

A (II) feladat majd fontos szerepet játszik az egyensúlyi pontot meghatározó algoritmusnál.

4. Az optimalizációs feladatok gazdasági szemléltetése

A következőkben kísérletet teszünk az optimalizációs feladatok egyféle gazdasági interpretálására. A célfüggvényeket közel „I-divergencia" (információ többlet) formára hozzuk. Sajnos az a tény, hogy a célfüggvények nem a matematikai értelembe vett pontos I-divergenciák, kétségessé teszik a közgazdasági szemléltetés korrektségét.

Elsőként a *Gale feladatot* és a (II) feladatot vizsgáljuk. A hivatkozás megkönnyítésére mindkét feladatot újra leírjuk.

Gale-feladat: Olyan $\mathbf{X} \geqq \mathbf{0}$ vásárlási mátrixot keressünk, amely a

$$\sum_{i=1}^{m} \xi_{ij} = \gamma_j, \quad (j = 1, \ldots, n) \tag{4.1}$$

áru egyensúlyi feltételt teljesíti és a

$$\sum_{i=1}^{m} \beta_i \log \frac{\beta_i}{\mathbf{a}_i \mathbf{x}_i} \tag{4.2}$$

célfüggvényt *minimalizálja.*

A β_i a C_i vásárló *tényleges* pénzkészlete ezt a jövedelem *a posteriori* eloszlásának tekintjük (itt feltehető, hogy 1-re normált eloszlás, minthogy $\sum_{i=1}^{n} \beta_i$ fix érték).

Az $\mathbf{a}_i \mathbf{x}_i$ érték a C_i vásárló haszna (elfogadott $\mathbf{X}$ vásárlás esetén) ezt *a priori* eloszlásnak tekintjük (itt az 1-re normáltságot nem tudjuk biztosítani).

Ekkor a (4.2) célfüggvény a két eloszlás I-divergencia eltérése. Így a *Gale feladat* úgy szemléltethető gazdaságilag, hogy olyan az áru egyensúlyt biztosító vásárlást keresünk, amely a keletkezett haszon eloszlásban az eredeti jövedelem eloszlást a lehető legjobban megközelíti.

(II) feladat: Olyan $\mathbf{p}>\mathbf{0}$ árvektort és $\mathbf{u}>\mathbf{0}$ haszonrátát keresünk, amely az

$$\alpha_{ij} \leqq \mu_i \pi_j, \quad (i = 1, \ldots, m;\ j = 1, \ldots, n), \tag{4.3}$$

egyenlőtlenségi feltételt és a

$$\sum_{j=1}^{n} \pi_j \gamma_j = \sum_{i=1}^{m} \beta_i \tag{4.4}$$

kereslet—kínálati egyensúlyi feltételt teljesíti és a

$$\sum_{i=1}^{m} \beta_i \log \frac{\beta_i}{\mu_i \beta_i} \tag{4.5}$$

célfüggvényt *maximalizálja.*

A β_i a C_i vásárló tényleges pénzkészlete, ezt a jövedelem *a posteriori* eloszlásának tekintjük (itt feltehető, hogy 1-re normált eloszlás, minthogy $\sum_{i=1}^{m} \beta_i$ adott érték).

A $\mu_i \beta_i$ érték a C_i vásárló haszna (elfogadott $\mathbf{u}$ haszonráta esetén) ezt *a priori* eloszlásnak tekintjük (itt az 1-re normáltságot nem tudjuk biztosítani).

Ekkor a (4.5) célfüggvény a két eloszlás I-divergencia eltérése.

Így a (II) feladat a következőképp szemléltethető gazdaságilag: Tegyük fel, hogy egy bank az árukért árakat ajánl, a fogyasztóknak pedig a pénzükért kamatot (haszonrátát). A C_i fogyasztó számára *elfogadható* az ajánlat, ha minden G_j árura (4.3) fennáll. Ugyanis ekkor semmilyen vásárlással sem tud magasabb hasznot elérni. A teljes pénz egyensúlynak (4.4)-nek pedig fenn kell állni. A bank célja, hogy passzív haszonszerzésnél az új jövedelem eloszlás kiegyenlítő legyen, azaz az eredetitől a lehető legjobban térjen el.

Míg a fogyasztó *aktív* tevékenységénél (vásárol) a szerzett jövedelem relatív nagy a meglevőhöz, addig *passzív* tevékenységnél (bankban helyezi el pénzét) a szerzett jövedelem relatív kicsi a meglevőhöz.

A két feladat kapcsolatára — figyelembevéve az 1.1. lemmát, a 2.1. lemmát és a 3.1. tételt — az alábbiakat nyerjük:

Létezik a *Gale feladatnak* optimális $\mathbf{X}^* \geqq \mathbf{0}$ megoldása — és létezik a (II) feladatnak $\mathbf{p}^*>\mathbf{0}$ és $\mathbf{u}^*>\mathbf{0}$ optimális megoldása és ez egyértelmű. Az optimális megoldáshoz tartozó célfüggvényértékek megegyeznek.

A feladatok megengedett megoldásai akkor és csak akkor optimálisak, ha

$$(\mu_i^* \pi_j^* - \alpha_{ij}) \xi_{ij}^* = 0, \quad (i = 1, \ldots, m;\ j = 1, \ldots, n), \tag{4.6}$$

és

$$\sum_{j=1}^{n} \pi_j^* \xi_{ij}^* = \beta_i, \quad (i = 1, \ldots, m), \tag{4.7}$$

azaz ha $\mathbf{p}^*$, $\mathbf{X}^*$, $\mathbf{u}^*$ egyensúlyi pont.

A következőkben az (I) feladatot vizsgáljuk. A hivatkozás megkönnyítésére a feladatot újra leírjuk:

(I) feladat: Olyan $\mathbf{p} \geqq \mathbf{0}$ árvektort és $\mathbf{X} \geqq \mathbf{0}$ vásárlási mátrixot keresünk, amelyek a

$$\sum_{j=1}^{n} \pi_j \xi_{ij} = \beta_i, \quad (i = 1, \ldots, m), \tag{4.8}$$

pénz egyensúlyi feltételt és a

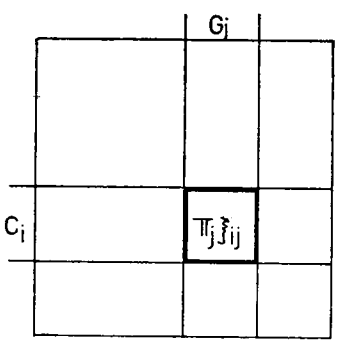
$$\sum_{i=1}^{m} \xi_{ij} = \gamma_j, \quad (j = 1, \ldots, n), \tag{4.9}$$

áru egyensúlyi feltételt teljesítik és a

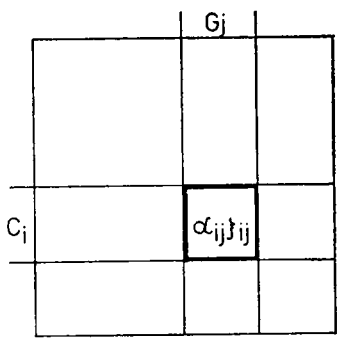
$$\sum_{i=1}^{m} \sum_{j=1}^{n} \pi_j \xi_{ij} \log \frac{\pi_j \xi_{ij}}{\alpha_{ij} \xi_{ij}} \tag{4.10}$$

célfüggvényt minimalizálják.

Tekintsük az $m \times n$-es vásárlási mátrix celláit.



4. ábra



5. ábra

Az $\mathbf{X} \geqq \mathbf{0}$ vásárlás és a $\mathbf{p} \geqq \mathbf{0}$ ár esetén az (i, j) cella ténylegesen $\pi_j \xi_{ij}$ értéket hoz. Ezt az értéket *a posteriori* eloszlásnak tekintjük. (Itt feltehető, hogy 1-re normált, minthogy $\sum_{i=1}^{m} \sum_{j=1}^{n} \pi_j \xi_{ij} = \sum_{i=1}^{m} \beta_i$ és $\sum_{i=1}^{m} \beta_i$ fix érték). (4. ábra).

Az $\mathbf{X} \geqq \mathbf{0}$ vásárlás esetén az árak tényleges értékével nem törődve az (i, j) cella a vásárlóknak $\alpha_{ij} \xi_{ij}$ értéket hoz. Ezt *a priori* eloszlásnak tekintjük. (Itt már az 1-re normáltság nem tehető fel.) (5. ábra).

Így az (I) feladat gazdaságilag úgy szemléltethető, hogy olyan a pénz egyensúlyt és az áruegyenleget biztosító árat és vásárlást keresünk, hogy a szubjektív cella érték eloszlás a tényleges cella értékeloszlást a lehető legjobban megközelítse.

A feladatra és az eredeti problémára — figyelembevéve az 1.1. lemmát és a 3.1. tételt — az alábbi kapcsolatot nyerjük.

Létezik a feladatnak $\mathbf{X}^* \geqq \mathbf{0}$ és $\mathbf{p}^* > \mathbf{0}$ optimális megoldása; $\mathbf{p}^*$ egyértelmű és pozitív. A feladat megengedett megoldásai akkor és csak akkor optimálisak, ha létezik $\mathbf{u}^* > \mathbf{0}$, úgy hogy

$$\alpha_{ij} \leqq \mu_i^* \pi_j^*, \quad (i = 1, \ldots, m;\ j = 1, \ldots, n),$$

és

$$(\mu_i^* \pi_j^* - \alpha_{ij}) \xi_{ij}^* = 0, \quad (i = 1, \ldots, m;\ j = 1, \ldots, n),$$

azaz ha $\mathbf{p}^*$, $\mathbf{X}^*$, $\mathbf{u}^*$ egyensúlyi pont.

5. Algoritmus az egyensúlyi pont meghatározására

Mint az előzőkben láttuk az egyensúlyi pont meghatározása az alábbi egyenlőtlenségi rendszer megoldása:

$$\xi_{ij} \geqq 0, \ \pi_j > 0, \ \mu_i > 0, \quad (i = 1, \ldots, m;\ j = 1, \ldots, n), \tag{5.1}$$

$$\sum_{j=1}^{n} \pi_j \xi_{ij} = \beta_i, \qquad (i = 1, \ldots, m), \tag{5.2}$$

$$(\mu_i \pi_j - \alpha_{ij}) \xi_{ij} = 0, \qquad (i = 1, \ldots, m;\ j = 1, \ldots, n), \tag{5.3}$$

$$\mu_i \pi_j \geqq \alpha_{ij}, \qquad (i = 1, \ldots, m;\ j = 1, \ldots, n), \tag{5.4}$$

$$\sum_{i=1}^{m} \xi_{ij} = \gamma_j, \qquad (j = 1, \ldots, n). \tag{5.5}$$

Felhasználjuk a (II) feladat eredményét: π_j, $\mu_i > 0$ akkor és csak akkor egyensúlyi, ha az

$$\alpha_{ij} \leqq \mu_i \pi_j \quad \text{és} \quad \sum_{j=1}^{n} \pi_j \gamma_j = \sum_{i=1}^{m} \beta_i$$

feltételek esetén az

$$\omega = \sum_{i=1}^{m} \beta_i \log \frac{1}{\mu_i} \tag{5.6}$$

maximális.

Bevezetve a (3.7) és (3.8)-ban már alkalmazott jelölést:

$$\beta = \sum_{i=1}^{m} \beta_i, \ \bar{\beta}_i = \frac{\beta_i}{\beta}, \ \bar{\alpha}_{ij} = \frac{\alpha_{ij} \gamma_j}{\beta}, \tag{5.7}$$

$$\bar{\pi}_j = \frac{\pi_j \gamma_j}{\beta}, \ \bar{\xi}_{ij} = \frac{\pi_j \xi_{ij}}{\beta}, \ \bar{\mu}_i = \mu_i, \tag{5.8}$$

az egyensúlyi feltételi rendszer az alábbi:

$$\xi_{ij} \geqq 0, \quad \bar{\pi}_j > 0, \quad \bar{\mu}_i > 0, \quad (i = 1, \ldots, m;\ j = 1, \ldots, n), \tag{5.9}$$

$$\sum_{j=1}^{n} \xi_{ij} = \beta_i, \quad (i = 1, \ldots, m), \tag{5.10}$$

$$(\bar{\mu}_i \bar{\pi}_j - \bar{\alpha}_{ij}) \xi_{ij} = 0, \quad (i = 1, \ldots, m;\ j = 1, \ldots, n), \tag{5.11}$$

$$\bar{\mu}_i \bar{\pi}_j \geqq \bar{\alpha}_{ij}, \quad (i = 1, \ldots, m;\ j = 1, \ldots, n), \tag{5.12}$$

$$\sum_{i=1}^{m} \xi_{ij} = \bar{\pi}_j, \quad (j = 1, \ldots, n). \tag{5.13}$$

A (II) feladat eredménye: $\bar{\pi}_j > 0$, $\bar{\mu}_i > 0$ akkor és csak akkor egyensúlyi, ha az

$$\bar{\alpha}_{ij} \leqq \bar{\mu}_i \bar{\pi}_j \quad \text{és} \quad \sum_{j=1}^{n} \bar{\pi}_j = 1 \tag{5.14}$$

feltételek esetén az

$$\bar{\omega} = \sum_{i=1}^{m} \beta_i \log \frac{1}{\bar{\mu}_i} \tag{5.15}$$

maximális.

Algoritmus: Legyenek $\bar{\mu}_i > 0$, $\bar{\pi}_j > 0$ olyanok, hogy az (5.14)-et teljesítik, (azaz a (II) feladat megengedett megoldásai). Jelölje $\bar{\omega}$ a hozzájuk tartozó (5.15)-ös célfüggvényértéket. Készítsük el az általános Kőnig—Hall modellt $\beta_1, \ldots, \beta_m$ kínálattal és $\bar{\pi}_1, \ldots, \bar{\pi}_n$ kereslettel. Legyen az (i, j) cella megengedett szállítású, ha $\bar{\alpha}_{ij} = \bar{\mu}_i \bar{\pi}_j$.

Oldjuk meg a *Kőnig—Hall feladatot* a címkézési technikával. Két eset lehetséges:

1° Ha a *Kőnig—Hall kereslet-kínálat modell* megoldható, akkor jelölve ξ_{ij}-vel a megoldást, akkor az (5.9)—(5.13) feltételek teljesülnek. Így megkaptuk az optimális megoldást és (5.8) transzformációkkal az eredeti feladatunk megoldását is.

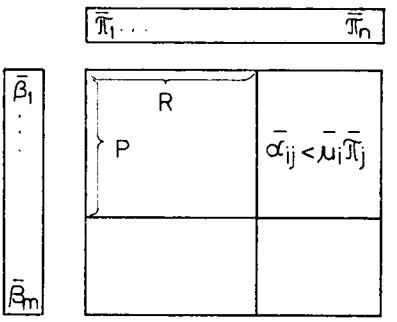
2° Ha a *Kőnig—Hall feladat* nem megoldható, akkor megfogunk adni egy új $\bar{\mu}_i' > 0$ és $\bar{\pi}_j' > 0$ az (5.14)-et kielégítő értékeket, amelyekhez tartozó (5.15)-ös célfüggvényérték $\bar{\omega}'$ határozottan nagyobb lesz mint $\bar{\omega}$. Ezt a következőképp tesszük:

Ha a *Kőnig—Hall feladat* nem megoldható, akkor a cimkézési technika kijelöli az i indexeknek P halmazát, és a j indexeknek R halmazát, melyekre:

$$\sum_{i \in P} \beta_i > \sum_{j \in R} \bar{\pi}_j \tag{5.16}$$

és

$$\bar{\alpha}_{ij} < \bar{\mu}_i \bar{\pi}_j, \quad \text{ha} \quad i \in P \quad \text{és} \quad j \notin R. \tag{5.17}$$



6. ábra

Jelöljük:

$$\delta = \min_{\substack{i \in P \\ j \notin R}} \frac{\bar{\mu}_i \bar{\pi}_j}{\bar{\alpha}_{ij}}, \quad \text{így} \quad \delta > 1. \tag{5.18}$$

$\left(\text{ha } \bar{\alpha}_{ij}=0, \text{ akkor a } \frac{\bar{\mu}_i \bar{\pi}_j}{\bar{\alpha}_{ij}} \text{ értéket } \infty\text{-nek tekintjük}\right)$.

Legyen ε tetszőleges szám, melyre $1 \leqq \varepsilon \leqq \delta$.
Legyenek:

$$\begin{aligned} \hat{\mu}_i &= \frac{\bar{\mu}_i}{\varepsilon}, \quad \text{ha} \quad i \in P, \\ \hat{\mu}_i &= \bar{\mu}_i, \quad \text{ha} \quad i \notin P, \\ \hat{\pi}_j &= \varepsilon \bar{\pi}_j, \quad \text{ha} \quad j \in R, \\ \hat{\pi}_j &= \bar{\pi}_j, \quad \text{ha} \quad j \notin R. \end{aligned} \tag{5.19}$$

Ezen $\hat{\mu}_i$, $\hat{\pi}_j$ értékekről egyszerűen beláthatók, hogy (5.14) egyenlőtlenségi feltételeit újra teljesítik, ugyanis:

Az $(i \in P; j \in R)$ esetben: $\hat{\mu}_i \hat{\pi}_j = \bar{\mu}_i \bar{\pi}_j \geqq \bar{\alpha}_{ij}$.

Az $(i \in P; j \notin R)$ esetben: $\hat{\mu}_i \hat{\pi}_j = \frac{\bar{\mu}_i}{\varepsilon} \bar{\pi}_j \geqq \bar{\alpha}_{ij}$, $\quad (\varepsilon \leqq \delta \text{ miatt})$.

Az $(i \notin P; j \in R)$ esetben: $\hat{\mu}_i \hat{\pi}_j = \bar{\mu}_i \bar{\pi}_j \cdot \varepsilon \geqq \bar{\alpha}_{ij}$, $\quad (\varepsilon \geqq 1 \text{ miatt})$.

Az $(i \notin P; j \notin R)$ esetben: $\hat{\mu}_i \hat{\pi}_j = \bar{\mu}_i \bar{\pi}_j \geqq \bar{\alpha}_{ij}$.

Ahhoz, hogy az új μ_i, π_j teljesítse (5.14) egyenlőségi feltételét is még egy normálás szükséges:

$$\begin{aligned} \bar{\mu}_i' &= \hat{\mu}_i \sum_{j=1}^{n} \hat{\pi}_j, \\ \bar{\pi}_j &= \frac{\hat{\pi}_j}{\sum_{j=1}^{n} \hat{\pi}_j}. \end{aligned} \tag{5.20}$$

Vizsgáljuk meg, hogy milyen ε esetén növekszik a legnagyobbra az új $\bar{\omega}'$ célfüggvény.

Az (5.15) szerint:

$$\bar{\omega}' = \sum_{i=1}^{m} \bar{\beta}_i \log \frac{1}{\bar{\mu}_i}. \tag{5.21}$$

A μ_i' értékeket — az (5.20) és (5.19) alatt adott módon — a $\bar{\mu}_i$, $\bar{\pi}_j$ értékekkel kifejezve, (5.21)-ből kapjuk, hogy

$$\bar{\omega}' = \sum_{i=1}^{m} \bar{\beta}_i \log \frac{1}{\bar{\mu}_i} + (\log \varepsilon) \sum_{i \in P} \bar{\beta}_i + \log \frac{1}{\varepsilon \sum\limits_{j \in R} \bar{\pi}_j + \sum\limits_{j \notin R} \bar{\pi}_j}. \tag{5.22}$$

A számolás egyszerűsítése érdekébe jelöljük:

$$p = \sum_{i \in P} \bar{\beta}_i \quad \text{és} \quad r = \sum_{j \in R} \bar{\pi}_j. \tag{5.23}$$

Az (5.16) miatt: $p > r$,

Ezen jelölést (5.22)-be átírva és átalakítva:

$$\bar{\omega}' = \bar{\omega} + \log \frac{\varepsilon^p}{\varepsilon r + 1 - r}. \tag{5.24}$$

Jelöljük az ω növekményét h-val:

$$h(\varepsilon) = \log \frac{\varepsilon^p}{\varepsilon r + 1 - r}. \tag{5.25}$$

A $h(\varepsilon)$ függvényre:

$$h(0) = -\infty,$$

$$h(1) = 0,$$

$$h(\varepsilon_0) = 0, \quad \left(\text{olyan } \varepsilon_0\text{-ra, melyre } r = \frac{\varepsilon_0^p - 1}{\varepsilon_0 - 1}\right)$$

$$h(\infty) = -\infty.$$

A $h(\varepsilon)$ függvény deriváltja:

$$\frac{dh(\varepsilon)}{d\varepsilon} = \frac{\varepsilon r(p-1) + p(1-r)}{\varepsilon(\varepsilon r + 1 - r)}. \tag{5.26}$$

Az (5.26) nevezője pozitív, minthogy $\varepsilon \geqq 1$, számlálója pedig akkor pozitív, ha

$$\varepsilon \leqq \frac{p(1-r)}{r(1-p)}$$

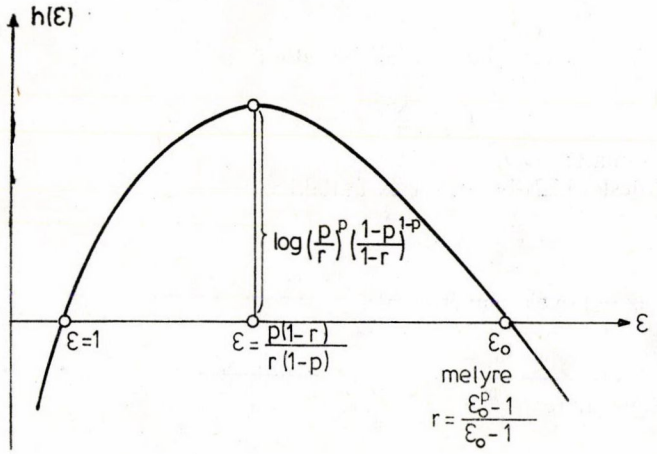
azaz $h(\varepsilon)$ függvény maximumát a $p(1-r)/r(1-p)$ helyen éri el; ez 1-nél nagyobb érték, minthogy $p > r$.

Hogy $\bar{\omega}'$ érték maximálisat növekedjék, az ε értéket

$$\varepsilon = \min\left(\delta, \frac{p(1-r)}{r(1-p)}\right) \tag{5.27}$$

módon kell választani, és ekkor a növekedés (5.25)-ből:

$$\log\left(\frac{p}{r}\right)^{p}\left(\frac{1-p}{1-r}\right)^{1-p}.$$



7. ábra

Összefoglalva az algoritmust a 2° esetben: A címkézési technika adja a P és R index halmazokat. Az (5.23)-mal kiszámoljuk a p és r értékeket. Az (5.18)-ban adott módon meghatározzuk δ értéket. Az (5.27) alatti módon meghatározzuk ε értékét. Az (5.19) illetve (5.20) alapján megadjuk az új $\bar{\mu}_i'$, $\bar{\pi}_j'$ értékeket.

Az eljárás konvergenciáját nem tudjuk igazolni, de számos computeren futott feladatnál minden esetben véges sok lépésben az eljárás — az 1° módon — véget ért.

IRODALOM

[1] EISENBERG, E. and GALE, D., "Consensus of subjective probabilities: The pari-mutuel method", *Ann. Math. Stat.* **30** (1959).
[2] GALE, D., *Theory of Linear Economic Models* (New York, McGraw-Hill, 1960).
[3] KLAFSZKY, E., "Geometriai programozás és néhány alkalmazása", *MTA-SZTAKI Tanulmányok* **8** (1973).

(Beérkezett: 1981. február 10.)

KLAFSZKY EMIL
NEHÉZIPARI MŰSZAKI EGYETEM MATEMATIKAI INTÉZET
3515 MISKOLC-EGYETEMVÁROS

THE DETERMINATION OF EQUILIBRIUM PRICES OF LINEAR EXCHANGE MODELS BY GEOMETRIC PROGRAMMING

E. KLAFSZKY

In the paper the equilibrium prices of linear exchange models (GALE [2]) is investigated.

The paper is divided into five sections.

In Section 1 the description of the model is given. It is shown — using the duality of linear programming — that the problem of the equilibrium price of an exchange model is equivalent to a linear complementary problem.

In Section 2 we prove the result of EISENBERG—GALE [1] — that the equilibrium point can be characterized by a mathematical programming problem due to GALE — but under an additional assumption that the equilibrium point exists. Our proof is very simple and depends only on the well-known aritmetic mean — geometric mean inequality. We refer to the above mentioned mathematical programming problem as the *Gale problem.*

In Section 3 we give two more mathematical programming problems (problems (I) and (II)) and prove — using the duality theorem of geometric programming [3] — that both of them also characterize the equilibrium point. From these the original result of EISENBERG—GALE can be derived very simply.

In Section 4 we try to give an economic interpretation of the three mentioned problems (the *Gale problem* and problem (I) and (II)) and we show that problem (II) and the *Gale problem* are in a dual relation.

In Section 5 we give a simple algorithm for the determination of the equilibrium point. This algorithm is based on the problem (II) and the *theorem of Kőnig—Hall* and is similar to the *Hungarian method* for transportation problems.

Alkalmazott Matematikai Lapok 7 (1981) 159—165

HÁROM ASZOCIÁLT LEGENDRE FÜGGVÉNY SZORZATÁNAK INTEGRÁLJA

KULLMANN LÁSZLÓ

Budapest

Gömbszimmetrikus áramlástani — és egyéb — problémák megoldása során előnyösen alkalmazható a *Galjerkin módszer*. Segítségével csökkenthető a numerikus számítás rácspontjainak száma. Forgásszimmetrikus esetben a *Galjerkin módszer elsőfajú aszociált Legendre függvények* alkalmazásához vezethet. Ha a probléma nemlineáris, akkor szükség van három *aszociált Legendre függvény* szorzatának integráljára a $-1 \leqq x \leqq 1$ intervallumon. Az integrál kiszámítására itt közölt módszer tette csak lehetővé, hogy elkerüljünk egy iterációt, amelyre a *Galjerkin módszer* alkalmazása nélkül eddig mindig szükség volt.

1. Bevezetés

A viszkózus folyadékok lamináris áramlását leíró *Navier—Stokes egyenlet*

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}\nabla)\mathbf{v} + \frac{1}{\varrho}\nabla p + \nabla U = \nu \Delta \mathbf{v} \tag{1.1}$$

és a kontinuitási egyenlet, ϱ=áll. mellett

$$\nabla \mathbf{v} = 0 \tag{1.2}$$

számos hidrodinamikai probléma megoldásának kiinduló egyenletrendszere. Az egyenletekben $\mathbf{v}$ a sebességvektor, p a nyomás, U a térerő potenciálja, ϱ a folyadék sűrűsége, ν a kinematikai viszkozitása és t az idő.

Az (1.1) egyenlet rotációját képezve p és U kiküszöbölhető:

$$\frac{\partial \boldsymbol{\Omega}}{\partial t} + (\mathbf{v}\nabla)\boldsymbol{\Omega} - (\boldsymbol{\Omega}\nabla)\mathbf{v} = \nu \Delta \boldsymbol{\Omega}, \tag{1.3}$$

ahol

$$\boldsymbol{\Omega} = \nabla \times \mathbf{v} \tag{1.4}$$

az örvényvektor. Előírva a kezdeti és peremfeltételeket, azokkal együtt az (1.2), (1.3), (1.4) egyenletrendszer meghatározza az időben változó áramképet. Részben, vagy teljesen falakkal határolt tartomány esetén a $\mathbf{v}$ sebességre vonatkozó tapadási feltétel azt jelenti, hogy a vizsgált tartományt határoló falak mentén a folyadék és a fal sebessége azonos.

Az $\boldsymbol{\Omega}$ örvényvektor fal melletti peremértékei azonban nem ismertek. Ezért egy — a véges differenciák módszerén alapuló — numerikus megoldás szükségszerűen kiegészítő feltételen alapuló iterációt tartalmaz. ($\boldsymbol{\Omega}$ becsült peremértékéből meghatározzuk az $\boldsymbol{\Omega}$ eloszlást az (1.3) egyenlettel és abból (1.4) segítségével a $\mathbf{v}$ eloszlást,

majd a kiegészítő feltétel segítségével Ω peremértékének egy jobb közelítését adjuk meg.) Ez az eljárás lassú, nagy a memóriaigénye és az időlépésenkénti iteráció konvergenciája további vizsgálatokat tesz szükségessé.

Az (1.2)—(1.4) egyenletrendszeren alapuló egész módszer legtöbb gondot okozó lépése az örvényvektor peremértékeinek helyes becslése, mint azt ROESNER [1] egy 1978-as nemzetközi konferencia nyitó előadásában megállapította.

Geofizikai, műszaki, meteorológiai problémák során gyakran célszerű a tartományt (r, ϑ, φ) gömbkoordinátákkal megadni, forgásszimmetrikus esetben $(\partial/\partial\varphi = 0)$ a keresett vektorterek a t, r, ϑ koordináták függvényei, r a sugarat, ϑ a forgástengellyel bezárt szöget (szélességet) jelöli. A *Galjerkin módszer* alkalmazásával tovább lehet csökkenteni a tér *numerikus számításba bevont* koordinátáinak számát és ez az előbbiek szerint jelentős számolási idő és memóriaigény csökkentést eredményezhet, figyelembe véve, hogy például az (1.2) és (1.4) egyenlet közönséges differenciálegyenletté egyszerűsödik.

A *Galjerkin módszert* abban a formájában alkalmazzuk, amikor a peremfeltételeket kielégítő függvények segítségével közelítjük a tartomány belsejében meghatározandó v függvényt, illetve annak v_j komponenseit

$$v_j = \sum_{i=1}^{N} \alpha_{ij} \varphi_{ij}(\vartheta) f_{ij}(t, r) \quad (j = 1, 2, 3).$$

Itt $\varphi_{ij}(\vartheta)$ lineárisan független függvényrendszer, α_{ij}-k a meghatározandó paraméterek, számuk N, $f_{ij}(r, t)$ a v_j sebességkomponens immár csak két koordinátától függő „része", amelyet továbbra is a véges differenciák módszerének alkalmazásával kívánunk meghatározni diszkrét (r, t) rácspontokban.

A gömbi koordinátákban felírt (1.2), (1.3) és (1.4) egyenletek szerkezetének és a gömbfelületeken előírt peremfeltételeknek a

$$\varphi_{ij}(\vartheta) = P^1_{i(j)}(\vartheta) \sin \vartheta \tag{1.5}$$

választás felel meg, $P^1_{i(j)}$ *elsőrendű, i-edfokú aszociált Legendre függvény*. Az $i(j)$ index paritása sebességkomponensenként különbözhet. Súlyfüggvényként ugyancsak *aszociált Legendre függvényeket* választunk. Azonos, *p-edrendű Legendre függvényekre* az ortogonalitási feltétel

$$\int_0^{\pi} P^p_i(\vartheta) P^p_{i'}(\vartheta) \sin \vartheta \, d\vartheta = \delta_{ii'} \frac{2}{2i+1} \frac{(i+p)!}{(i-p)!},$$

ahol $\delta_{ii'}$, a *Kronecker-delta*.

Mivel azonban az (1.3) egyenlet nemlineáris, az α_{ij} paraméterek meghatározásakor a másodrendű tagoknak megfelelően

$$J = \int_0^{\pi} P^p_l(\vartheta) P^q_m(\vartheta) P^r_n(\vartheta) \sin \vartheta \, d\vartheta \tag{1.6}$$

típusú integrálokat kell kiszámítani. A *Legendre függvényekre* vonatkozó rekurziós formulákat és differenciálási szabályokat alkalmazva, a felső index értéke 0, 1 vagy 2, az alsó index pedig a meghatározandó α_{ij} paraméterek N számától függ. Sem az általam ismert, *aszociált Legendre függvényeket* tárgyaló könyvekben, mint például ERDÉLYI [2], FARKAS [3], JAHNKE [4], LENSE [5], sem számos, az alkalmazásokat ismertető könyvben, cikkben nem találtam megfelelő integrálképletet.

2. A J integrál kiszámítása

Természetesen a parciális differenciálegyenletek alkalmazása során másutt is előállítható a megoldás gömbfüggvények segítségével (a *Legendre függvények* speciális gömbfüggvények). A kvantummechanikában elemi részecskecsoportok impulzusnyomatékának meghatározása három harmonikus gömbfüggvény szorzatának integráljához vezet. Az

$$I = \int_0^{2\pi} \int_0^{\pi} Y_{lp}(\vartheta, \varphi) Y_{mq}(\vartheta, \varphi) Y_{ns}(\vartheta, \varphi) \sin \vartheta \, d\vartheta \, d\varphi$$

integrált kell ott kiszámítani (ALDER [6], EDMONDS [7], FLÜGGE [8]). Az alkalmazott módszert nem részletezve, a továbbiakhoz szükséges végeredmény:

$$I = \left[\frac{(2l+1)(2m+1)(2n+1)}{4\pi}\right]^{1/2} \begin{pmatrix} l & m & n \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} l & m & n \\ p & q & s \end{pmatrix} \tag{2.1}$$

(lásd pl. [7], 4.6.3 jelű integrálképletét). A zárójelek között álló 2-szer 3 számot *Wigner-féle* $3-j$ *szimbólumnak* hívják, ezekre a későbbiekben visszatérek.

Mivel a *harmonikus gömbfüggvények* és az *aszociált Legendre függvények* kapcsolatban vannak egymással, éspedig (pl. [7] 2.5.29 képlet)

$$Y_{lp}(\vartheta, \varphi) = (-1)^p \left[\frac{(2l+1)(l-p)!}{4\pi(l+p)!}\right]^{1/2} P_l^p(\vartheta) e^{ip\varphi}, \tag{2.2}$$

a (2.1) eredmény segít a keresett (1.6) integrál kiszámításában. A megfelelő indexekkel (2.2)-t (2.1)-be behelyettesítve és az $s=-r$ jelölést bevezetve

$$I = \int_0^{2\pi} \int_0^{\pi} \frac{(-1)^{p+q-r}}{4\pi} \left[\frac{(2l+1)(2m+1)(2n+1)(l-p)!(m-q)!(n+r)!}{4\pi(l+p)!(m+q)!(n-r)!}\right]^{1/2} \cdot \tag{2.3}$$

$$\cdot P_l^p P_m^q P_n^{-r} e^{i(p+q-r)} \sin \vartheta \, d\vartheta \, d\varphi.$$

Továbbá kihasználva, hogy ([7], 2.5.18 képlet)

$$P_n^{-r} = (-1)^r \frac{(n-r)!}{(n+r)!} P_n^r, \tag{2.4}$$

végül az I integrál így írható

$$I = \frac{(-1)^{p+q}}{4\pi} \left[\frac{(2l+1)(2m+1)(2n+1)}{4\pi}\right]^{1/2} \left[\frac{(l-p)!(m-q)!(n-r)!}{(l+p)!(m+q)!(n+r)!}\right]^{1/2} \cdot \tag{2.5}$$

$$\cdot \int_0^{2\pi} e^{i(p+q-r)\varphi} d\varphi \cdot J.$$

Mivel p, q, r egész, így $p+q-r$ is egész. Ha $p+q-r=0$, akkor

$$\int_0^{2\pi} e^{i(p+q-r)\varphi} \, d\varphi = \int_0^{2\pi} d\varphi = 2\pi. \tag{2.6}$$

Ha $p+q-r=k\neq 0$, akkor

$$\int_0^{2\pi} e^{i(p+q-r)\varphi}\,d\varphi = \int_0^{2\pi} e^{ik\varphi}\,d\varphi = 0. \tag{2.7}$$

a) eset: $p+q=r$

Először vizsgáljuk azt az esetet, amikor $p+q-r=0$. Ekkor a (2.5) összefüggés bal oldalát a (2.1) képlettel kifejezve, a jobboldalra a (2.6) egyenlőséget beírva és figyelembe véve, hogy $p+q-r=0$ miatt $(-1)^{p+q}=(-1)^r$ és $s=-r$, a J határozott integrál értéke a következő képletből számítható

$$J=\int_0^{\pi} P_l^p P_m^q P_n^r \sin\vartheta\,d\vartheta = 2(-1)^r\cdot \tag{2.8}$$

$$\cdot\left[\frac{(l+p)!(m+q)!(n+r)!}{(l-p)!(m-q)!(n-r)!}\right]^{1/2}\begin{pmatrix} l & m & n\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix} l & m & n\\ p & q & r\end{pmatrix}.$$

A másik esetben, ha $p+q-r\neq 0$, akkor a (2.5) egyenlőség jobb oldalának egyik tényezője zérus, de ugyanekkor a (2.1) képletben a második $3-j$ szimbólum is zérus értékű a *Wigner-féle szimbólum* alább felsorolt egyik tulajdonsága miatt, így $I=0$. Ekkor tehát a J értékére rendezés után 0/0 alakú kifejezés adódik.

Az (1.2)—(1.4) egyenletrendszerre vonatkozó probléma megoldása során a p, q és r alkalmas választásával mindig sikerült elérni, hogy $p+q-r=0$ legyen, azaz a (2.8) képlet alkalmazható volt. Ha más alkalmazás során ez nem lenne elérhető, akkor [6] (159. o., 17. képlet) alapján a következőből kell kiindulni.

$$P_m^q(\vartheta)P_n^r(\vartheta)=\sum_{k=|m-n|}^{m+n} D(k,m,q,n,r)P_k^s(\vartheta),\quad s=|r-q|, \tag{2.9}$$

azaz két *aszociált Legendre függvény* szorzata előállítható ilyenek lineáris kombinációjaként. A D együttható ismét kapcsolatban van a *Wigner-féle* $3-j$ *szimbólumokkal* ([6], 174. o.)

$$D(k,m,q,n,r)=(-1)^{\frac{1}{2}(q+r+|q-r|)}\left[\frac{(k-|q-r|)!(m+q)!(n+r)!}{(k+|q-r|)!(m-q)!(n-r)!}\right]^{1/2}\cdot \tag{2.10}$$

$$\cdot(2k+1)\begin{pmatrix} m & n & k\\ q & -r & r-q\end{pmatrix}\begin{pmatrix} m & n & k\\ 0 & 0 & 0\end{pmatrix}.$$

A (2.9) képletet $P_l^p(\vartheta)\sin\vartheta$-val szorozva és 0-tól π-ig integrálva a J integrál előállítható

$$J=\int_0^{\pi} P_l^p P_m^q P_n^r \sin\vartheta\,d\vartheta = \sum_{k=|m-n|}^{m+n} D(k,m,q,n,r)\int_0^{\pi} P_l^p(\vartheta)P_k^s(\vartheta)\sin\vartheta\,d\vartheta. \tag{2.11}$$

Ha $p>0$ és $r=p+q$, akkor $s=|r-q|=p$ és az azonos rendű *aszociált Legendre függvények* ortogonalitása miatt

$$J=D(l,m,q,n,r)\frac{2}{2l+1}\frac{(l+p)!}{(l-p)!}.$$

Ezt a D együtthatók (2.10) képletével egybevetve ($k=l$, $p=r-q$) a korábbi (2.8) integrálképletet kapjuk.

Mint látható, két különböző gondolatmenettel is el lehet jutni a keresett határozott integrál értékéhez, az első kézenfekvőbb a *harmonikus gömbfüggvények* és *aszociált Legendre függvények* közötti rokonság alapján. A második gondolatmenet viszont lehetővé teszi, hogy $r \neq p+q$ esetén is ki lehessen számítani a kérdéses integrált.

b) eset: $p+q \neq r$

A pozitív p, q, r számok közül legyen $r \geqq p$, $r \geqq q$, ekkor a (2.9) képletet egymás után kétszer alkalmazva

$$P_n^r P_l^p = \sum_{k'=|l-n|}^{l+n} D(k', n, r, l, p) P_{k'}^{r-p}$$

és

$$P_{k'}^{r-p} P_m^q = \sum_{k=|m-k'|}^{m+k'} D(k, k', r-p, m, q) P_k^{|r-p-q|}.$$

A J integrál kiszámításához tehát az $s=|r-p-q|$ jelöléssel

$$\int_0^{\pi} P_k^s(\vartheta) \sin\vartheta \, d\vartheta = \int_{-1}^{1} P_k^s(x)\, dx, \quad x = \cos\vartheta \tag{2.12}$$

alakú integrálok lineáris kombinációját kell kiszámítani. Ez utóbbi rekurziós képletekkel lehetséges, ha a $P_k^s(x)$ függvényt x hatványsoraként állítjuk elő ([4], 115. oldal, első képlet). A hatványsor j-edik tagjának alakja

$$\beta(k, s, j)(1-x^2)^{\frac{s}{2}} x^{k-s-2j} \quad (j = 0, 1, 2, \ldots),$$

$\beta(k, s, j)$ konstans. Végül tehát

$$\int_{-1}^{1} (1-x^2)^{\frac{s}{2}} x^b \, dx$$

típusú integrálokat kell kiszámítani. Ennek a binomiális differenciálnak az integrálásához rekurziós formulák állnak rendelkezésre (pl. BRONSTEJN [9], 452. old. 273. képlet, vagy KORN [10], 841. old. 273. képlet). Azok alapján

$$\int_{-1}^{1} (1-x^2)^{\frac{s}{2}} x^b \, dx = \begin{cases} \dfrac{s}{b+s+1} \displaystyle\int_{-1}^{1} (1-x^2)^{\frac{s}{2}-1} x^b \, dx & \begin{pmatrix}\text{Az első tényező}\\ \text{kitevője csökken}\end{pmatrix} \\ \dfrac{b-1}{b+s+1} \displaystyle\int_{-1}^{1} (1-x^2)^{\frac{s}{2}} x^{b-2} \, dx & \begin{pmatrix}\text{A második tényező}\\ \text{kitevője csökken}\end{pmatrix}. \end{cases}$$

A felsorolt képletek sorozatos alkalmazásával tehát kiszámítható a J integrál értéke $r \neq p+q$ esetben is.

3. A $3-j$ szimbólumok

Végül a *Wigner-féle* $3-j$ *szimbólumokat* illetően az irodalomra (ROTENBERG [11], 1.5 képlet) utalok

$$\begin{pmatrix} l & m & n \\ p & q & r \end{pmatrix} = (-1)^{l-m+r} \cdot$$

$$\cdot \left[\frac{(j-2l)!(j-2m)!(j-2n)!(l+p)!(l-p)!(m+q)!(m-q)!(n-r)!(n+r)!}{(j+1)!} \right]^{1/2} \cdot$$

$$\cdot \sum_k \frac{(-1)^k}{k!(l+m-n-k)!(l-p-k)!(m+q-k)!(n-m+p+k)!(n-l-q+k)!}, \tag{3.1}$$

ahol $j=l+m+n$. Ha l, m, n nem elégíti ki a háromszög egyenlőtlenséget, vagy $p+q-r \neq 0$, akkor a $3-j$ szimbólum értéke zérus. Az összegzést azokra a tagokra kell elvégezni, amelyekben nem fordul elő negatív szám faktoriálisa, az összeget jelölje: S.

$$\begin{pmatrix} l & m & n \\ 0 & 0 & 0 \end{pmatrix} = (-1)^{j/2} \left[\frac{(j-2l)!(j-2m)!(j-2n)!}{(j+1)!} \right]^{1/2} \frac{(j/2)!}{(j/2-l)!(j/2-m)!(j/2-n)!}. \tag{3.2}$$

Ha j páratlan szám, akkor ez a $3-j$ szimbólum is 0 értékű.

A (3.1) és (3.2) képletet behelyettesítve a J integrál (2.8) értékébe — S jelöli a (3.1) képletbeli összeget — végül azt kapjuk, hogy

$$J = \int_0^\pi P_l^p(\vartheta) P_m^q(\vartheta) P_n^r(\vartheta) \sin\vartheta \, d\vartheta = 2(-1)^{l-m+\frac{j}{2}+2r} \cdot$$

$$\cdot \frac{(j-2l)!(j-2m)!(j-2n)!(l+p)!(m+q)!(n+r)!}{(j/2-l)!(j/2-m)!(j/2-n)!(j+1)!} S.$$

A bevezetőben vázolt áramlástani probléma numerikus megoldására készített FORTRAN program ezt a képletet használja.

A faktoriálisok értékének kiszámítása nehézséget jelent. 40! 10^{48} nagyságrendű, 60! már 10^{82} nagyságrendű, tehát a számítógép számábrázolása korlátot szab a kiszámítható faktoriális nagyságának. Több szerző ([6], [11]) ezért azt javasolja, hogy minden egész számot előbb prímszámok szorzataként kell előállítani és a szám helyett csak a prímszámok hatványkitevőit kell tárolni. Szorzáskor tehát a hatványkitevők összegződnek. Összegzés előtt pedig a legnagyobb közös osztót ki kell emelni és azután visszaírni az összeg tagjait szokásos alakba, majd összegezni. A részleteket illetően az irodalomra utalok [11].

A *Galjerkin-módszer* alkalmazása — ehhez a J integrál numerikus kiszámítása szükséges — azt eredményezi, hogy a bevezetőben vázolt probléma iteráció nélkül megoldható, mert az (1.4) egyenlet közönséges differenciálegyenletté egyszerűsödik

és zárt alakban megoldható. Így nincs szükség az Ω vektor peremértékének explicit megadására. Az iteráció elkerülésének két előnye is van. Egyrészt nem kell konvergencia-vizsgálatot végezni, másrészt az Ω vektor peremértékére nem kell előírást tenni. Egy ilyen pótlólagos előírás ugyanis döntően befolyásolja a numerikus számítás eredményét, mint azt ISRAELI [12] és WU [13] behatóan vizsgálták.

IRODALOM

[1] ROESNER, K. G., "Numerical calculation of hydrodynamic stability problems with time-dependent boundary conditions", *Proc. Sixth. Int. Conf. on Numerical Methods in Fluid Dynamics*, Springer, Berlin (1978) 1—25.

[2] ERDÉLYI, A., MAGNUS, W., OBERHETTINGER, F. and TRICOMI, F. G., *Higher Transcendental Functions, Vol.* **1**. (McGraw-Hill, New York, 1953).

[3] FARKAS, M., *Speciális függvények műszaki-fizikai alkalmazásokkal* (Műszaki Kiadó, Budapest, 1964).

[4] JAHNKE, EMDE und LÖSCH, *Tafeln höherer Funktionen* (Teubner, Stuttgart, 1960).

[5] LENSE, J., *Kugelfunktionen* (Akademische Verlagsgesellschaft, Leipzig, 1950).

[6] ALDER, B., FERNBACH, S. and ROTENBERG, M., *Methods in Computational Physics, Vol. 2. Quantum Mechanics* (Academic Press, New York, 1963).

[7] EDMONDS, A. R., *Angular Momentum in Quantum Mechanics* (Princeton Univ. Press, Princeton, 1957).

[8] FLÜGGE, S., *Handbuch der Physik, Band XXXIX. Bau der Atomkerne* (Springer, Berlin, 1957).

[9] BRONSTEJN, I. N. és SZEMENGYAJEV, K. A., *Matematikai Zsebkönyv* (Műszaki Könyvkiadó, Budapest, 1974).

[10] KORN, G. A. és KORN, T. M., *Matematikai Kézikönyv Műszakiaknak* (Műszaki Könyvkiadó, Budapest, 1975).

[11] ROTENBERG, M., BIVINS, R., METROPOLIS, N. and WOOTEN, J. K., *The 3−j and 6−j symbols* (Crossby Lockwood & Son LTD, London, 1959).

[12] ISRAELI, M., "On the evaluation of iteration parameters for the boundary vorticity", *Studies in Applied Mathematics* **51** (1972) 67—71.

[13] WU, J. C., "Numerical boundary conditions for viscous flow problems", *AIAA-Journal* **14** (1976) 1042—1049.

(Beérkezett: 1981. március 2.)

KULLMANN LÁSZLÓ
BME VÍZGÉPEK TANSZÉK
1521 BUDAPEST, STOCZEK U. 2.

THE INTEGRAL OF THE PRODUCT OF THREE ASSOCIATED LEGENDRE FUNCTIONS

L. KULLMANN

When solving fluid dynamics and other problems with spherical geometry, the *Galerkin-method* may be used with advantage. Among others the number of grid points can be reduced by the aid of the method. In cases with rotational symmetry, the *Galerkin-method* can lead to the application of *associated Legendre functions of the first kind.* When the problem is nonlinear it is necessary to integrate the product of three *associated Legendre functions* above the interval $-1 \leqq x \leqq 1$. It is the algorithm of eveluation of the integral described here, which made possible to avoid an iteration procedure necessary unless using the *Galerkin-method.*

Alkalmazott Matematikai Lapok 7 (1981) 167—174

KORLÁTOK AZ ENERGIAINTEGRÁL SZÁMÁRA

ECSEDI ISTVÁN

Miskolc

E tanulmány tárgya közönséges, másodrendű, lineáris, inhomogén, változó együtthatójú differenciálegyenlet rendszerre vonatkozik.

Alsó és felső korlátot ismertet a differenciálegyenlet rendszer megoldásához rendelt energiaintegrál számára. A korlátok bizonyításai döntően a *Schwarz egyenlőtlenség* alkalmazásán alapulnak.

1. Bevezetés

Tekintsük az alábbi kerületérték feladatot:

$$-\frac{d}{dx}\left[\mathbf{P}(x)\frac{d\mathbf{u}}{dx}\right]+\mathbf{Q}(x)\mathbf{u}=\mathbf{f}(x), \quad x_1<x<x_2, \tag{1.1}$$

$$-\mathbf{P}_1\frac{d\mathbf{u}}{dx}+\mathbf{M}_1\mathbf{u}=\mathbf{0}, \quad x=x_1, \tag{1.2}$$

$$\mathbf{P}_2\frac{d\mathbf{u}}{dx}+\mathbf{M}_2\mathbf{u}=\mathbf{0}, \quad x=x_2. \tag{1.3}$$

A fenti egyenletekben: $\mathbf{u}=\mathbf{u}(x)$ az x $(x_1\leqq x\leqq x_2)$ valós változó függvényének tekintett n dimenziós oszlopvektor (vektor-skalár függvény); $\mathbf{P}=\mathbf{P}(x)$ az $x_1<x<x_2$ intervallumban legalább egyszer folytonosan differenciálható, az $x_1\leqq x\leqq x_2$ intervallumban folytonos n-edrendű pozitív definit, szimmetrikus négyzetes mátrix (mátrix-skalár függvény); $\mathbf{Q}=\mathbf{Q}(x)$ az $x_1\leqq x\leqq x_2$ intervallumban folytonos n-edrendű pozitív definit szimmetrikus négyzetes mátrix (mátrix-skalár függvény); $\mathbf{P}_1=\mathbf{P}(x_1)$, $\mathbf{P}_2=\mathbf{P}(x_2)$; $\mathbf{f}=\mathbf{f}(x)$ az $x_1<x<x_2$ intervallumban folytonos n dimenziós oszlopvektor (vektor-skalár függvény); $\mathbf{M}_1$, $\mathbf{M}_2$ n-edrendű szimmetrikus pozitív definit négyzetes mátrixok; x_1, x_2 $(x_1<x_2)$ adott valós állandók.

A tanulmány a mátrix számítás szokásos jelöléseit alkalmazza, így a kitevőben szereplő „T" a transzponálás, „-1" pedig az invertálás műveleti jele.

Az (1.1), (1.2), (1.3) kerületérték feladat $\mathbf{u}=\mathbf{u}(x)$ $(x_1\leqq x\leqq x_2)$ megoldásához rendelt energiaintegrál alatt az alábbi előírással meghatározott valós számot értjük [1]:

$$E=-\int_{x_1}^{x_2}\mathbf{u}^T\frac{d}{dx}\left(\mathbf{P}\frac{d\mathbf{u}}{dx}\right)dx+\int_{x_1}^{x_2}\mathbf{u}^T\mathbf{Q}\mathbf{u}\,dx. \tag{1.4}$$

Elöljáróban bebizonyítjuk, hogy az

$$Lv = -\frac{d}{dx}\left[\mathbf{P}(x)\frac{d\mathbf{v}}{dx}\right] + \mathbf{Q}(x)\mathbf{v} \tag{1.5}$$

előírással értelmezett operátor pozitív, lineáris, szimmetrikus, ha az L operátor D_L értelmezési tartományát a szükséges differenciálhatósági feltételeken felül az

$$-\mathbf{P}_1\frac{d\mathbf{v}}{dx} + \mathbf{M}_1\mathbf{v} = \mathbf{0}, \quad x = x_1, \tag{1.6}$$

$$\mathbf{P}_2\frac{d\mathbf{v}}{dx} + \mathbf{M}_2\mathbf{v} = \mathbf{0}, \quad x = x_2 \tag{1.7}$$

homogén kerületi feltételt kielégítő $\mathbf{v}=\mathbf{v}(x)$ $(x_1 \leqq x \leqq x_2)$ vektor-skalár függvények alkotják, megjegyezvén, hogy a $\mathbf{v}=\mathbf{v}(x)$ és $\mathbf{w}=\mathbf{w}(x)$ $[\mathbf{v}, \mathbf{w} \in D_L]$ vektor-skalár függvények $(\mathbf{v}, \mathbf{w})$ skaláris szorzatát a

$$(\mathbf{v}, \mathbf{w}) = \int_{x_1}^{x_2} \mathbf{v}^T(\xi)\mathbf{w}(\xi)\,d\xi \tag{1.8}$$

formulával értelmezzük.

Az L operátor lineáris volta nyilvánvaló, szimmetriája pedig az alábbi egyenletekből következik:

$$(\mathbf{w}, L\mathbf{v}) = -\int_{x_1}^{x_2} \mathbf{w}^T \frac{d}{dx}\left(\mathbf{P}\frac{d\mathbf{v}}{dx}\right)dx + \int_{x_1}^{x_2} \mathbf{w}^T\mathbf{Q}\mathbf{v}\,dx = \int_{x_1}^{x_2} \left(\frac{d\mathbf{w}}{dx}\right)^T \mathbf{P}\frac{d\mathbf{v}}{dx}\,dx + \tag{1.9}$$

$$+ \int_{x_1}^{x_2} \mathbf{w}^T\mathbf{Q}\mathbf{v}\,dx + \mathbf{w}^T(x_1)\mathbf{M}_1\mathbf{v}(x_1) + \mathbf{w}^T(x_2)\mathbf{M}_2\mathbf{v}(x_2),$$

$$(\mathbf{v}, L\mathbf{w}) = -\int_{x_1}^{x_2} \mathbf{v}^T \frac{d}{dx}\mathbf{P}\frac{d\mathbf{w}}{dx} + \int_{x_1}^{x_2} \mathbf{v}^T\mathbf{Q}\mathbf{w}\,dx = \int_{x_1}^{x_2} \left(\frac{d\mathbf{v}}{dx}\right)^T \mathbf{P}\frac{d\mathbf{w}}{dx}\,dx + \tag{1.10}$$

$$+ \int_{x_1}^{x_2} \mathbf{v}^T\mathbf{Q}\mathbf{w}\,dx + \mathbf{v}^T(x_1)\mathbf{M}_1\mathbf{w}(x_1) + \mathbf{v}^T(x_2)\mathbf{M}_2\mathbf{w}(x_2).$$

Az (1.9) és (1.10) formulák levezetésénél alkalmaztuk a parciális integrálás szabályát és a $\mathbf{v}, \mathbf{w} \in D_L$ függvényekkel kapcsolatos (1.6), (1.7) kerületi feltételeket.

Az (1.9) és (1.10) egyenletek összevetéséből közvetlenül adódik, hogy

$$(L\mathbf{v}, \mathbf{w}) = (L\mathbf{w}, \mathbf{v}), \tag{1.11}$$

vagyis az L operátor szimmetrikus.

Legyen az (1.9) formulában $\mathbf{v}=\mathbf{w}$

$$(L\mathbf{v}, \mathbf{v}) = \int_{x_1}^{x_2} \left(\frac{d\mathbf{v}}{dx}\right)^T \mathbf{P}\frac{d\mathbf{v}}{dx}\,dx + \tag{1.12}$$

$$+ \int_{x_1}^{x_2} \mathbf{v}^T\mathbf{Q}\mathbf{v}\,dx + \mathbf{v}^T(x_1)\mathbf{M}_1\mathbf{v}(x_1) + \mathbf{v}^T(x_2)\mathbf{M}_2\mathbf{v}(x_2).$$

Az (1.12) formulából kiolvasható, hogy L pozitív operátor, hiszen

$$(Lv, v) \geqq 0 \tag{1.13}$$

és

$$(Lv, v) = 0 \tag{1.14}$$

csak akkor, ha $v=v(x)$ az $x_1 \leqq x \leqq x_2$ intervallumban az azonos zérus vektor.

Az (1.13) egyenlőtlenség következménye, hogy az E energiaintegrál nem negatív:

$$E \geqq 0. \tag{1.15}$$

Könnyen kimutatható, hogy E csak akkor zérus, ha

$$\int_{x_1}^{x_2} \mathbf{f}^T(\xi)\mathbf{f}(\xi)\, d\xi = 0. \tag{1.16}$$

Az (1.1) egyenlet és az (1.4) formula kombinálásával kapjuk az (1.17) formulát:

$$E = \int_{x_1}^{x_2} \mathbf{f}^T(\xi)\mathbf{u}(\xi)\, d\xi. \tag{1.17}$$

E tanulmány elsődleges célja az (1.4) formula által definiált E energiaintegrál becslését lehetővé tevő alsó és felső korlátok előállítása.

2. Alsó korlát

2.1. TÉTEL. Az (1.1), (1.2), (1.3) egyenletek által kijelölt kerületérték feladat $\mathbf{u}=\mathbf{u}(x)$ $x_1 \leqq x \leqq x_2$ megoldásához rendelt E energia integrál számára az alábbi alsó korlát adható:

$$E \geqq \frac{A}{B}, \tag{2.1}$$

$$A = \left(\int_{x_1}^{x_2} \mathbf{f}^T \mathbf{a}\, dx \right)^2, \tag{2.2}$$

$$B = \int_{x_1}^{x_2} \left(\frac{d\mathbf{a}}{dx} \right)^T \mathbf{P} \frac{d\mathbf{a}}{dx}\, dx + \tag{2.3}$$

$$+ \int_{x_1}^{x_2} \mathbf{a}^T \mathbf{Q} \mathbf{a}\, dx + \mathbf{a}^T(x_1)\mathbf{M}_1\mathbf{a}(x_1) + \mathbf{a}^T(x_2)\mathbf{M}_2\mathbf{a}(x_2),$$

ahol $\mathbf{a}=\mathbf{a}(x)$ az $x_1 \leqq x \leqq x_2$ intervallumban folytonos és az $x_1 < x < x_2$ intervallumban szakaszonként folytonosan differenciálható az azonos zérus vektortól különböző egyébként tetszőleges n-dimenziós vektor (vektor-skalár függvény).

Bizonyítás: Tekintsük az n-dimenziós $\mathbf{a}=\mathbf{a}(x)$ és $\mathbf{b}=\mathbf{b}(x)$ vektor-skalár függvények alábbi előírással képzett $[\mathbf{a}, \mathbf{b}]$ skaláris szorzatát:

$$[\mathbf{a}, \mathbf{b}] = \int_{x_1}^{x_2} \left(\frac{d\mathbf{a}}{dx}\right)^T \mathbf{P} \frac{d\mathbf{b}}{dx}\, dx + \tag{2.4}$$

$$+ \int_{x_1}^{x_2} \mathbf{a}^T \mathbf{Q} \mathbf{b}\, dx + \mathbf{a}^T(x_1)\mathbf{M}_1\mathbf{b}(x_1) + \mathbf{a}^T(x_2)\mathbf{M}_2\mathbf{b}(x_2).$$

Az $\mathbf{a}=\mathbf{a}(x)$ és $\mathbf{b}=\mathbf{b}(x)$ vektor-skalár függvényekkel kapcsolatban egyelőre csak azt tételezzük fel, hogy a (2.4) formulában szereplő integrálok léteznek és véges értékűek legyenek.

Az értelmezés alapján nyilvánvaló, hogy

$$[\mathbf{a}, \mathbf{b}] = [\mathbf{b}, \mathbf{a}], \tag{2.5}$$

$$[\mathbf{a}, \mathbf{a}] \geqq 0 \tag{2.6}$$

és $[\mathbf{a}, \mathbf{a}]=0$ csak akkor, ha

$$\int_{x_1}^{x_2} \mathbf{a}^T(\xi)\mathbf{a}(\xi)\, d\xi = 0. \tag{2.7}$$

Schwarz egyenlőtlensége szerint

$$[\mathbf{a}, \mathbf{a}][\mathbf{b}, \mathbf{b}] \geqq ([\mathbf{a}, \mathbf{b}])^2. \tag{2.8}$$

A (2.8) egyenlőtlenségben alkalmazzuk a $\mathbf{b}=\mathbf{u}(x)$ helyettesítést és tételezzük fel, hogy $\mathbf{a}=\mathbf{a}(x)$ az $x_1 \leqq x \leqq x_2$ intervallumban folytonos, az $x_1 < x < x_2$ intervallumban pedig szakaszonként folytonosan differenciálható az azonosan zérus vektortól különböző vektor-skalár függvényt jelöl.

Az $[\mathbf{a}, \mathbf{u}]$ skaláris szorzat átalakításával nyerjük a (2.9) formulát:

$$[\mathbf{a}, \mathbf{u}] = \int_{x_1}^{x_2} \left(\frac{d\mathbf{a}}{dx}\right)^T \mathbf{P} \frac{d\mathbf{u}}{dx}\, dx + \int_{x_1}^{x_2} \mathbf{a}^T \mathbf{Q}\mathbf{u}\, dx + \mathbf{a}^T(x_1)\mathbf{M}_1\mathbf{u}(x_1) + \tag{2.9}$$

$$+ \mathbf{a}^T(x_2)\mathbf{M}_2\mathbf{u}(x_2) = \left[\mathbf{a}^T \mathbf{P} \frac{d\mathbf{u}}{dx}\right]_{x_1}^{x_2} + \int_{x_1}^{x_2} \mathbf{a}^T \left(-\frac{d}{dx}\mathbf{P}\frac{d\mathbf{u}}{dx} + \mathbf{Q}\mathbf{u}\right) dx +$$

$$+ \mathbf{a}^T(x_1)\mathbf{M}_1\mathbf{u}(x_1) + \mathbf{a}^T(x_2)\mathbf{M}_2\mathbf{u}(x_2) = \int_{x_1}^{x_2} \mathbf{a}^T(x)\mathbf{f}(x)\, dx.$$

Ez utóbbi eredmény és a nyilvánvaló

$$E = [\mathbf{u}, \mathbf{u}] = \int_{x_1}^{x_2} \mathbf{f}^T(x)\mathbf{u}(x)\, dx \tag{2.10}$$

egyenlőség alkalmazásával a (2.8) *Schwarz-féle egyenlőtlenségből* a bizonyítandó (2.1) egyenlőtlenségi relációt nyerjük.

3. Felső korlát

3.1. TÉTEL. Az (1.1), (1.2), (1.3) egyenletek által kijelölt kerületérték feladat $\mathbf{u}=\mathbf{u}(x)$ $x_1 \leqq x \leqq x_2$ megoldásához rendelt E energiaintegrál számára az alábbi felső korlát adható:

$$E \leqq \int_{x_1}^{x_2} \mathbf{d}_1^T \mathbf{P} \mathbf{d}_1 \, dx + \int_{x_1}^{x_2} \mathbf{d}_2^T \mathbf{Q} \mathbf{d}_2 dx + \tag{3.1}$$
$$+\mathbf{d}_1^T(x_1)\mathbf{P}_1\mathbf{M}_1^{-1}\mathbf{P}_1\mathbf{d}_1(x_1)+\mathbf{d}_1^T(x_2)\mathbf{P}_2\mathbf{M}_2^{-1}\mathbf{P}_2\mathbf{d}_1(x_2),$$

ahol $\mathbf{d}_1=\mathbf{d}_1(x)$ az $x_1 \leqq x \leqq x_2$ intervallumban folytonos, az $x_1<x<x_2$ intervallumban pedig folytonosan differenciálható, $\mathbf{d}_2=\mathbf{d}_2(x)$ pedig az $x_1<x<x_2$ intervallumban folytonos olyan vektor-skalár függvényeket jelölnek, melyek az alábbi egyenletet kielégítik:

$$-\frac{d}{dx}(\mathbf{P}\mathbf{d}_1)+\mathbf{Q}\mathbf{d}_2 = \mathbf{f}, \quad x_1 < x < x_2. \tag{3.2}$$

Bizonyítás: Tekintsük a

$$\mathbf{c}^T = \mathbf{c}^T(x) = [\mathbf{c}_1^T(x), \mathbf{c}_2^T(x)]$$

és a

$$\mathbf{d}^T = \mathbf{d}^T(x) = [\mathbf{d}_1^T(x), \mathbf{d}_2^T(x)]$$

$2n$ dimenziós vektorokból felépülő *Euklédeszi teret,* ahol $\mathbf{c}_1$, $\mathbf{c}_2$, illetve $\mathbf{d}_1$, $\mathbf{d}_2$ n-dimenziós vektorok.

A fentiekben definiált $2n$ dimenziós *Euklédeszi térben* a vektor összeadást és a skalárral való szorzást a szokásos módon értelmezzük. A $\mathbf{c}$ és $\mathbf{d}$ vektorok $\{\mathbf{c}, \mathbf{d}\}$ skaláris szorzatát pedig a következő előírással definiáljuk:

$$\{\mathbf{c}, \mathbf{d}\} = \int_{x_1}^{x_2} [\mathbf{c}_1^T \mathbf{P} \mathbf{d}_1 + \mathbf{c}_2^T \mathbf{Q} \mathbf{d}_2] \, dx + \tag{3.3}$$
$$+\mathbf{c}_1^T(x_1)\mathbf{P}_1\mathbf{M}_1^{-1}\mathbf{P}_1\mathbf{d}_1(x_1)+\mathbf{c}_1^T(x_2)\mathbf{P}_2\mathbf{M}_2^{-1}\mathbf{P}_2\mathbf{d}_1(x_2).$$

Az értelmezés alapján belátható, hogy

$$\{\mathbf{c}, \mathbf{d}\} = \{\mathbf{d}, \mathbf{c}\}, \tag{3.4}$$

$$\{\mathbf{c}, \mathbf{c}\} \geqq 0 \tag{3.5}$$

és $\{\mathbf{c}, \mathbf{c}\}$ csak akkor zérus, ha

$$\int_{x_1}^{x_2} \mathbf{c}_1^T \mathbf{c}_1 \, dx + \int_{x_1}^{x_2} \mathbf{c}_2^T \mathbf{c}_2 \, dx = 0,$$

továbbá

$$\{\mathbf{c}, \mathbf{c}\}\{\mathbf{d}, \mathbf{d}\} \geqq (\{\mathbf{c}, \mathbf{d}\})^2. \tag{3.6}$$

A (3.6) *Schwarz-féle* egyenlőtlenségi relációban legyen

$$\mathbf{c}_1 = \frac{d\mathbf{u}}{dx}, \quad \mathbf{c}_2 = \mathbf{u}$$

a $\mathbf{d}_1 = \mathbf{d}_1(x)$ és a $\mathbf{d}_2 = \mathbf{d}_2(x)$ n-dimenziós vektorokat pedig úgy választjuk meg, hogy az alábbi egyenlet fennálljon:

$$-\frac{d}{dx}(\mathbf{P}\mathbf{d}_1) + \mathbf{Q}\mathbf{d}_2 = \mathbf{f}, \quad x_1 < x < x_2. \tag{3.7}$$

Könnyen verifikálható, hogy ez esetben

$$E = \{\mathbf{c}, \mathbf{c}\}. \tag{3.8}$$

Következőkben a $\{\mathbf{c}, \mathbf{d}\}$ skaláris szorzat kifejtésével foglalkozunk:

$$\{\mathbf{c}, \mathbf{d}\} = \int_{x_1}^{x_2} \left(\frac{d\mathbf{u}}{dx}\right)^T \mathbf{P}\mathbf{d}_1 \, dx + \tag{3.9}$$

$$+ \int_{x_1}^{x_2} \mathbf{u}^T \mathbf{Q}\mathbf{d}_2 \, dx + \left(\frac{d\mathbf{u}}{dx}\right)_{x=x_1}^T \mathbf{P}_1 \mathbf{M}_1^{-1} \cdot \mathbf{P}_1 \mathbf{d}_1(x_1) + \left(\frac{d\mathbf{u}}{dx}\right)_{x=x_2} \mathbf{P}_2 \mathbf{M}_2^{-1} \mathbf{P}_2 \mathbf{d}_1(x_2) =$$

$$= \int_{x_1}^{x_2} \mathbf{u}^T \left[-\frac{d}{dx}(\mathbf{P}\mathbf{d}_1) + \mathbf{Q}\mathbf{d}_2\right] dx + [\mathbf{u}^T \mathbf{P}\mathbf{d}_1]_{x_1}^{x_2} + \mathbf{u}^T(x_1)\mathbf{P}_1\mathbf{d}_1(x_1) -$$

$$-\mathbf{u}^T(x_2)\mathbf{P}_2\mathbf{d}_1(x_2) = \int_{x_1}^{x_2} \mathbf{f}^T \mathbf{u} \, dx = E.$$

A levezetés során a parciális integrálás szabályát alkalmaztuk és felhasználtuk, hogy az $\mathbf{u} = \mathbf{u}(x)$ függvény kielégíti az (1.2), (1.3) peremfeltételeket.

A (3.6) egyenlőtlenség és a (3.8), (3.9) egyenletek kombinálásával közvetlenül a bizonyítandó (3.1) egyenlőtlenségi relációt nyerjük.

4. Néhány egyszerű szerkezetű korlát

4.1. Legyen

$$\mathbf{F} = \int_{x_1}^{x_2} \mathbf{f}(\xi) \, d\xi, \tag{4.1}$$

$$\mathbf{q} = \int_{x_1}^{x_2} \mathbf{Q}(\xi) \, d\xi. \tag{4.2}$$

Tetszőleges zérus vektortól különböző állandó $\mathbf{b}$ vektorral fennáll az

$$E \geqq \frac{\mathbf{b}^T \mathbf{F} \mathbf{F}^T \mathbf{b}}{\mathbf{b}^T(\mathbf{q} + \mathbf{M}_1 + \mathbf{M}_2)\mathbf{b}} \tag{4.3}$$

egyenlőtlenségi reláció.

4.2. Legyen

$$\mathbf{k} = \int_{x_1}^{x_2} \xi \mathbf{f}(\xi)\, d\xi, \tag{4.4}$$

$$\mathbf{p} = \int_{x_1}^{x_2} \mathbf{P}(\xi)\, d\xi, \tag{4.5}$$

$$\mathbf{R} = \int_{x_1}^{x_2} \xi^2 \mathbf{Q}(\xi)\, d\xi. \tag{4.6}$$

Tetszőleges zérus vektortól különböző állandó $\mathbf{c}$ vektorral fennáll az

$$E \geqq \frac{\mathbf{c}^T \mathbf{k}\mathbf{k}^T \mathbf{c}}{\mathbf{c}^T (\mathbf{R} + \mathbf{p} + x_1^2 \mathbf{M}_1 + x_2^2 \mathbf{M}_2)\mathbf{c}} \tag{4.7}$$

egyenlőtlenségi reláció.

A (4.3) és (4.7) alsó korlátok bizonyítása közvetlenül adódik a (2.1) egyenlőtlenségi relációból, ha azt az alábbi alakú $\mathbf{a}=\mathbf{a}(x)$, $x_1 \leqq x \leqq x_2$ vektorokra alkalmazzuk:

$$\mathbf{a}(x) = \mathbf{b}, \quad (\mathbf{b} = \text{állandó vektor } \mathbf{b} \neq \mathbf{o}), \tag{4.8}$$

$$\mathbf{a}(x) = \mathbf{c}x, \tag{4.9}$$

($\mathbf{c}$=állandó vektor, $\mathbf{c} \neq \mathbf{o}$).

4.3. Fennáll az

$$E \leqq \int_{x_1}^{x_2} \mathbf{f}^T(\xi)[\mathbf{Q}(\xi)]^{-1}\mathbf{f}(\xi)\, d\xi \tag{4.10}$$

egyenlőtlenségi reláció.

4.4. Legyen

$$\mathbf{h} = \mathbf{h}(x) = -\int_{x_1}^{x} \mathbf{f}(\xi)\, d\xi. \tag{4.11}$$

Fennáll az

$$E \leqq \int_{x_1}^{x_2} \mathbf{h}^T(\xi)\mathbf{P}^{-1}(\xi)\mathbf{h}(\xi)\, d\xi + \mathbf{h}^T(x_2)\mathbf{M}_2^{-1}\mathbf{h}(x_2) \tag{4.12}$$

egyenlőtlenségi reláció.

A (4.10) és (4.12) felső korlátok bizonyítása a (3.1) egyenlőtlenségi relációból a következő helyettesítések alkalmazásával nyerhető:

$$\begin{aligned} \mathbf{d}_1 &= \mathbf{d}_1(x) = \mathbf{o}, \quad x_1 \leqq x \leqq x_2, \\ \mathbf{d}_2 &= \mathbf{d}_2(x) = [\mathbf{Q}(x)]^{-1}\mathbf{f}(x), \quad x_1 \leqq x \leqq x_2, \end{aligned} \tag{4.13}$$

$$\begin{aligned} \mathbf{d}_1 &= \mathbf{d}_1(x) = [\mathbf{P}(x)]^{-1}\mathbf{h}(x), \quad x_1 \leqq x \leqq x_2, \\ \mathbf{d}_2 &= \mathbf{d}_2(x) = \mathbf{o}, \qquad\qquad\quad x_1 \leqq x \leqq x_2. \end{aligned} \tag{4.14}$$

IRODALOM

[1] Михлин, С. Г., *Вариационные методы в математической физике.* (Изд. Наука, Москва, 1970).

(Beérkezett: 1980. december 15.)

ECSEDI ISTVÁN
NME MECHANIKAI TANSZÉK
3515 MISKOLC-EGYETEMVÁROS

BOUNDS FOR THE ENERGY INTEGRAL

I. Ecsedi

This paper deals with systems of second order differential eqations. Upper and lower bounds are derived for the energy integral wich is connected with solution of the system of differential eqations. The formulas (3.1), (4.10), (4.12) give upper bounds and the formulas (2.1), (4.3), (4.7) give lower bounds for the energy integral.

Alkalmazott Matematikai Lapok 7 (1981) 175—180

TÖBBSZÖRÖS GYÖKPÁROKKAL RENDELKEZŐ VALÓS EGYÜTTHATÓS POLINOMOK FAKTORIZÁLÁSA

VARGA GYULA

Budapest

A cikk egy, a *Newton—Raphson-módszeren* alapuló eljárást ad meg — a *Bairstow-eljárás* általánosításaként — ismert multiplicitású többszörös gyökpárokkal rendelkező valós együtthatós polinomok faktorizálására. Az eljárás konvergenciája másodrendű.

1. Bevezetés

Valós együtthatós polinomok faktorizálása a numerikus matematika egyik alapvető feladata. A probléma sokrétű, az egyes módszereknek különféle változatai vannak kidolgozva egyszeres és többszörös, valós és komplex gyökökkel rendelkező polinomok faktorizálására, pontosabban az illető típusú gyökökhöz tartozó gyöktényezők leválasztására (l. pl. [1], [2], [3]). Egyszeres konjugált komplex vagy legfeljebb kétszeres valós gyökpárokhoz tartozó valós másodfokú tényezők kiszámítására és leválasztására szolgál a *Newton—Raphson-eljárás Bairstow-féle változata,* amely a keresett másodfokú tényező együtthatóit úgy számítja ki, hogy a polinomot a közelítő másodfokú tényezővel elosztva, a kapott elsőfokú osztási maradékot (amelynek együtthatói éppen a közelítő másodfokú tényező együtthatóitól függnek) nullázza. Az eljárás egyik általánosítása alkalmas arra, hogy ha a polinom valamennyi valós másodfokú tényezőjének jó közelítése áll rendelkezésünkre, az összes másodfokú tényezőt egyidejűleg számítsa ki, elkerülve ezzel a hibák felhalmozódását [4]. A jelen dolgozatban a *Bairstow-féle változatnak* egy más irányú általánosítását közöljük, amely ismert multiplicitású többszörös valós másodfokú tényezők kiszámítására és leválasztására alkalmas, tehát valós együtthatós polinomok többszörös komplex gyökei is egyszerűen, valós úton számíthatók ki vele, s egyidejűleg a hányadospolinomot is megadja. Az eljárás konvergenciája, megegyezően a *Bairstow-féle változatéval,* másodrendű. A továbbiakban ismertetjük az eljárást, bebizonyítjuk másodrendű konvergenciáját, és a megjegyzésekben útmutatást adunk alkalmazásához.

2. A polinomfaktorizálási algoritmus

Legyen $a(x)$ n-edfokú valós együtthatós polinom ($n>0$), legyenek p_1 és p_2 az $a(x)$ polinom egy ismert multiplicitású valós másodfokú tényezőjének közelítő együtthatói, m a multiplicitás ($0<m<n/2$), és vezessük be a $\mathbf{p}=[p_1, p_2]^T$ jelölést.

Osszuk el az $a(x)$ polinomot m-szeres közelítő másodfokú tényezőjével. Az osztást az alábbi polinom-azonosság segítségével írhatjuk fel:

$$a(x) \equiv b(\mathbf{p}, x)\cdot(x^2+p_1x+p_2)^m+c(\mathbf{p}, x), \tag{2.1}$$

ahol $b(\mathbf{p}, x)$ a hányadospolinom, a

$$c(\mathbf{p}, x) = \sum_{j=1}^{2m} c_j(\mathbf{p}) x^{2m-j} \tag{2.2}$$

$2m-1$-edfokú polinom pedig az osztási maradék. Az osztás elvégzéséhez az osztó $h_1, \ldots, h_{2m+1}$ együtthatói a polinomiális együtthatókra vonatkozó

$$\begin{aligned} &h_i = \delta_{i,1} \ (\delta \text{ a } \textit{Kronecker-szimbólum}, \ i \leqq 2m+1); \\ &k = 1, m; \ j = 2k+1, \ 2k, \ \ldots, \ 1; \\ &h_j = h_j + h_{j-1} p_1 + h_{j-2} p_2 \end{aligned} \tag{2.3}$$

rekurzióval egyszerűen előállíthatók. Deriválva (2.1)-et p_1 szerint, az alábbi azonosságot kapjuk:

$$0 \equiv mxb(\mathbf{p}, x) \cdot (x^2 + p_1 x + p_2)^{m-1} + \frac{\partial b(\mathbf{p}, x)}{\partial p_1} (x^2 + p_1 x + p_2)^m + \frac{\partial c(\mathbf{p}, x)}{\partial p_1}. \tag{2.4}$$

Mivel a közelítő másodfokú tényező gyökei egyúttal a $2m-1$-edfokú derivált maradékpolinomnak is $m-1$-szeres gyökei (ez a (2.4) azonosságból látható), azért a p_1 szerint derivált maradékpolinom szükségképpen

$$\frac{\partial c(\mathbf{p}, x)}{\partial p_1} = \big(r_1(\mathbf{p}) x + r_2(\mathbf{p})\big) \cdot (x^2 + p_1 x + p_2)^{m-1} \tag{2.5}$$

alakban írható fel. Ezt (2.4)-be behelyettesítve és $(x^2+p_1x+p_2)^{m-1}$-nel egyszerűsítve az alábbi azonosságot kapjuk:

$$0 \equiv mxb(\mathbf{p}, x) + \frac{\partial b(\mathbf{p}, x)}{\partial p_1} (x^2 + p_1 x + p_2) + r_1(\mathbf{p}) x + r_2(\mathbf{p}). \tag{2.6}$$

Látható, hogy $r_1(\mathbf{p})$ és $r_2(\mathbf{p})$ a (2.1) polinomosztás hányadospolinomja $-mx$-szeresének $x^2+p_1x+p_2$-vel való elosztásakor adódó elsőfokú maradékpolinom együtthatói. A (2.1)-et p_2 szerint deriválva, hasonló meggondolás alapján kapjuk, hogy a p_2 szerint derivált maradékpolinom szükségképpen az alábbi alakú:

$$\frac{\partial c(\mathbf{p}, x)}{\partial p_2} = \big(s_1(\mathbf{p}) x + s_2(\mathbf{p})\big) \cdot (x^2 + p_1 x + p_2)^{m-1}. \tag{2.7}$$

A p_1 szerinti deriválthoz hasonlóan behelyettesítés és ugyancsak $(x^2+p_1x+p_2)^{m-1}$-nel való egyszerűsítés révén adódik a

$$0 \equiv mb(\mathbf{p}, x) + \frac{\partial b(\mathbf{p}, x)}{\partial p_2} (x^2 + p_1 x + p_2) + s_1(\mathbf{p}) x + s_2(\mathbf{p}) \tag{2.8}$$

azonosság, amelyből látható, hogy $s_1(\mathbf{p})$ és $s_2(\mathbf{p})$ a (2.1) polinomosztás hányadospolinomja $-m$-szeresének $x^2+p_1x+p_2$-vel való elosztásakor adódó elsőfokú maradékpolinom együtthatói.

A fenti polinomosztásokra és átalakításokra azért volt szükségünk, mert az $\mathbf{f}(\mathbf{p})=[c_1(\mathbf{p}), c_2(\mathbf{p})]^T=\mathbf{0}$ nemlineáris egyenletrendszert a *Newton—Raphson-féle iterációs eljárás* segítségével akarjuk megoldani ($c_1(\mathbf{p})$ és $c_2(\mathbf{p})$ a (2.2)-ből adódik a 2 legmagasabbfokú tag együtthatójaként), és az iterációhoz szükséges

$$J(\mathbf{p})=\left[\frac{\partial c_j(\mathbf{p})}{\partial p_k}\right] \quad (j,k=1,2) \tag{2.9}$$

Jacobi-féle mátrix elemeit a 2 utóbbi polinomosztás maradékaiból akarjuk meghatározni.

Deriválva (2.2)-t p_k $(k=1,2)$ szerint az alábbi egyenlőséget kapjuk:

$$\frac{\partial c(\mathbf{p},x)}{\partial p_k}=\sum_{j=1}^{2m}\frac{\partial c_j(\mathbf{p})}{\partial p_k}x^{2m-j} \quad (k=1,2). \tag{2.10}$$

A (2.5) és (2.10), illetve a (2.7) és (2.10), két-két legmagasabbfokú tagja együtthatóinak összehasonlításából adódnak a

$$\begin{aligned}&\frac{\partial c_1(\mathbf{p})}{\partial p_1}=r_1(\mathbf{p}), \quad \frac{\partial c_2(\mathbf{p})}{\partial p_1}=r_2(\mathbf{p})+(m-1)p_1r_1(\mathbf{p}),\\ &\frac{\partial c_1(\mathbf{p})}{\partial p_2}=s_1(\mathbf{p}), \quad \frac{\partial c_2(\mathbf{p})}{\partial p_2}=s_2(\mathbf{p})+(m-1)p_1s_1(\mathbf{p})\end{aligned} \tag{2.11}$$

egyenlőségek az első parciális deriváltakra. Ha a (2.8) azonosságot megszorozzuk x-szel, és figyelembe vesszük az

$$x(s_1(\mathbf{p})x+s_2(\mathbf{p}))\equiv s_1(\mathbf{p})(x^2+p_1x+p_2)+(s_2(\mathbf{p})-p_1s_1(\mathbf{p}))x-p_2s_1(\mathbf{p}) \tag{2.12}$$

azonosságot, akkor láthatjuk, hogy a (2.6) és (2.8) polinomosztások maradékai között az alábbi összefüggések állnak fenn:

$$r_1(\mathbf{p})=s_2(\mathbf{p})-p_1s_1(\mathbf{p}), \quad r_2(\mathbf{p})=-p_2s_1(\mathbf{p}), \tag{2.13}$$

tehát a parciális deriváltak meghatározásához voltaképpen csak egy polinomosztást kell végeznünk, a (2.8)-at.

Most már, kiindulva egy alkalmas $\mathbf{p}^{(0)}$ kezdeti közelítésből, alkalmazhatjuk a *Newton—Raphson-iterációt* az $\mathbf{f}(\mathbf{p})=\mathbf{0}$ nemlineáris egyenletrendszer megoldására:

$$\mathbf{p}^{(i+1)}=\mathbf{p}^{(i)}-J^{-1}(\mathbf{p}^{(i)})\mathbf{f}(\mathbf{p}^{(i)}) \quad (i=0,1,\ldots). \tag{2.14}$$

Ha az iteráció konvergens, eredményeképpen a (2.2) osztási maradéknak látszólag csupán az első két tagja válik zérussá, de érvényes a következő

2.1. TÉTEL. Ha az alábbi 3 feltétel teljesül:

$$1)\quad \exists\mathbf{p}^*\colon \left.\frac{d^ja(x)}{dx^j}\right|_{x=x^*_{1,2}}=0 \quad (j=0,1,\ldots,m-1),$$

ahol $$x^*_{1,2}=(-p_1^*\pm\sqrt{d^*})/2 \quad \text{és} \quad d^*=p_1^{*2}-4p_2^*;$$

$$2)\quad |b(\mathbf{p}^{(i)},x^{(i)}_{1,2})|>M>0 \quad (i=0,1,\ldots),$$

ahol $x_{1,2}^{(i)} = (-p_1^{(i)} \pm \sqrt{d^{(i)}})/2$ és $d^{(i)} = p_1^{(i)2} - 4p_2^{(i)}$;

3) $|d^{(i)}| > K > 0 \quad (i = 0, 1, \ldots)$,

akkor a $\mathbf{p}^{(i)}$ sorozat tart egy olyan $\mathbf{p}^*$-hoz, amelyre $a(x) = b(\mathbf{p}^*, x) \cdot (x^2 + p_1^* x + p_2^*)^m$. A konvergencia másodrendű.

Bizonyítás: Írjuk fel a (2.1) polinom-azonosságot $\mathbf{p}^*$ segítségével:

$$a(x) \equiv b(\mathbf{p}^*, x) \cdot (x^2 + p_1^* x + p_2^*)^m + c(\mathbf{p}^*, x). \tag{2.15}$$

Ezt az azonosságot x szerint $m-1$-szer deriválva, és a deriváltakat az $x = x_{1,2}^*$ helyeken véve adódnak a

$$\left.\frac{d^j a(x)}{dx^j}\right|_{x=x_{1,2}^*} = \left.\frac{\partial^j c(\mathbf{p}^*, x)}{\partial x^j}\right|_{x=x_{1,2}^*} = 0 \quad (j = 0, 1, \ldots, m-1) \tag{2.16}$$

egyenlőségek, amelyekből következik, hogy mint x polinomja $c(\mathbf{p}^*, x) \equiv 0$. Ugyanis a $c(\mathbf{p}^*, x)$ függvényt az $x = x_{1|}^*$ hely körül *Taylor-sorba* fejtve, és az együtthatókat g_i-vel $(i = 0, \ldots, 2m-1)$ jelölve, (2.16) miatt $g_0 = \ldots = g_{m-1} = 0$, tehát

$$c(\mathbf{p}^*, x) = \sum_{j=m}^{2m-1} g_j \frac{(x - x_1^*)^j}{j!}.$$

Minthogy továbbá a

$$\frac{c(\mathbf{p}^*, x)}{(x - x_1^*)^m} = \sum_{j=0}^{m-1} g_{j+m} \frac{(x - x_1^*)^j}{(j+m)!}$$

$m-1$-edfokú polinomnak a helyettesítési értéke és első $m-1$ deriváltja az $x = x_2^* (\neq x_1^*)$ helyen ugyancsak (2.16) miatt 0, és az ezt kifejező homogén lineáris egyenletrendszer mátrixa háromszögű, csupa 0-tól különböző átlós elemekkel, azért csak a triviális megoldás létezik, vagyis $g_m = \ldots = g_{2m-1} = 0$, és minthogy $\mathbf{f}(\mathbf{p}^*) = [c_1(\mathbf{p}^*), c_2(\mathbf{p}^*)]^T = 0$, ezért $\mathbf{p}^*$ a (2.14) iteráció fixpontja.

Érvényes tehát az

$$a(x) = b(\mathbf{p}^*, x) \cdot (x^2 + p_1^* x + p_2^*)^m \tag{2.17}$$

felbontás.

Végül (2.11) és (2.13) segítségével felírhatjuk a *Jacobi-mátrix* determinánsát:

$$\det^{(i)} = p_2^{(i)} s_1^2(\mathbf{p}^{(i)}) + s_2^2(\mathbf{p}^{(i)}) - p_1^{(i)} s_1(\mathbf{p}^{(i)}) s_2(\mathbf{p}^{(i)}). \tag{2.18}$$

Minthogy tényezőkre bontva

$$\det^{(i)} = \prod_{k=1}^{2} (s_1(\mathbf{p}^{(i)}) x_k^{(i)} + s_2(\mathbf{p}^{(i)})), \tag{2.19}$$

továbbá $\det^{(i)} \to \det^*$, és $x_{1,2}^{(i)} \to x_{1,2}^*$, azért $\det^* \neq 0$, mert különben $k=1$-re vagy $k=2$-re (2.8) miatt $b(\mathbf{p}^*, x_k^*) = 0$ lenne, ami 2) miatt lehetetlen, tehát a (2.14) iterációs eljárás konvergenciája másodrendű.

3. Megjegyzések

1. A tételben szereplő első feltétel azt fejezi ki, hogy az $x_{1,2}^*$ gyökök és velük együtt az $(x-x_1^*)(x-x_2^*)=x^2+p_1^*x+p_2^*$ másodfokú tényező multiplicitása *legalább* m legyen, a második feltétel pedig azt, hogy a multiplicitás *legfeljebb* m legyen. Emiatt az első 2 feltétel egyben szükséges is.

2. Az eljárás használata főként többszörös konjugált komplex gyökpárokhoz tartozó valós másodfokú tényezők hatványainak leválasztására célszerű, hiszen többszörös valós gyökökhöz tartozó magasabbfokú gyöktényezők egyszerűbb algoritmus segítségével is leválaszthatók a polinomról (l. pl. [3]). A tételben szereplő 3) feltétel, amely kizárja azt, hogy a másodfokú tényezőnek kétszeres gyöke legyen, ugyancsak emiatt nem jelent megszorítást.

3. Az eljárás FORTRAN szubrutinja a (2.3) rekurziót tartalmazza a közelítő tényező együtthatóinak előállítására, továbbá a (2.1) és (2.8) polinomosztásokat és a (2.13) átalakítást a függvényérték és a Jacobi-mátrix elemeinek a kiszámítására, végül a (2.14) iterációt az $\mathbf{f}(\mathbf{p})=\mathbf{0}$ nemlineáris egyenletrendszer megoldására. A szubrutin és paraméterei a következők:

SUBROUTINE MUCOFA (N, M, A, U, B, C, P, EPS, ITER, IER)

N	a polinom fokszáma
M	a keresett tényező ismert multiplicitása
A	a polinom tömbje csökkenő fokszám szerint rendezve $(n+1)$
U	a maradékpolinom tömbje $(2m+1)$
B	munkatömb a közelítő tényező együtthatói számára $(2m+1)$
C	a hányadospolinom tömbje $(n-2m+1)$
P	együtthatóvektor (2)
EPS	a megengedett abszolút eltérés
ITER	a megengedett (I)/végrehajtott (O) iterációs lépések száma
IER	hibakód: 0 OK
	1 lépésszámtúllépés
	−1 szingularitás.

4. Ha a keresett másodfokú tényező multiplicitását nem ismerjük, akkor a főprogramban a szubrutint ismételten hívhatjuk növekvő m értékekkel, amíg a hibakód −1-nek adódik. Ez ugyanis annak a jele, hogy a valódi multiplicitás nagyobb, mint az m paraméter értéke. Ilyenkor a szubrutin újrahívása az előző hívás eredményeként kapott **p**-vel, mint kezdő értékkel is történhet.

5. Az eljárás szubrutinjának kipróbálása az MTA CDC 3300-as gépén történt.

IRODALOM

[1] RALSTON, A., *Bevezetés a numerikus analízisbe* (Műszaki Könyvkiadó, Budapest, 1969).
[2] Березин, И. С. и Жидков, Н. П., *Методы вычислений* (Москва, Наука, 1966).
[3] VARGA, GY., „Többszörös valós gyökökkel rendelkező valós együtthatós polinomok faktorizálása", *Alk. Mat. Lapok* (megjelenés alatt).
[4] VARGA, GY., „Polinomfaktorizálás másodfokú tényezőkre", *MTA SZK Közlemények* **5** (1969).

(Beérkezett: 1980. október 29.)

VARGA GYULA
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1250 BUDAPEST, URI U. 49.

FACTORIZATION OF REAL POLYNOMIALS WITH MULTIPLE PAIRS OF ROOTS

GY. VARGA

The paper gives a procedure based on the *Newton—Raphson method* as the generalization of the *Bairstow-procedure* for factorization of real polynomials, which have multiple pairs of roots of known multiplicity. The convergence of the procedure is quadratic.

Alkalmazott Matematikai Lapok 7 (1981) 181—183

ÉSZREVÉTELEK A BAIRSTOW-MÓDSZER KONVERGENCIÁJÁVAL KAPCSOLATBAN

VARGA GYULA

Budapest

A cikk a *Bairstow-módszer* alkalmazhatóságával és konvergenciaviszonyaival foglalkozik, külö nös tekintettel a kétszeres valós gyökökkel rendelkező polinomokra.

1. Bevezetés

A *Bairstow-módszerről* írt cikkében BOYD [1] figyelemreméltó megállapításokat tesz és bizonyít be a módszer konvergenciájával, illetve a bizonyos esetekben fellépő divergenciájával kapcsolatban, amelynek megszüntetésére egyszerű javaslatot is tesz. Cikke elején tett némelyik megállapítása azonban helyesbítésre illetve pontosításra szorul. Az egyik az a kitétele, hogy a módszer valós együtthatós polinomok valós másodfokú tényezőinek a meghatározására szolgál, a másik pedig az, hogy a módszer konvergenciája egyszeres gyökök és kétszeres valós gyökök esetén másodrendű.

Az első megállapítást röviden kiigazítjuk, megemlítve azt a tényt, hogy a módszer levezetése során sehol sem használjuk ki azt a feltételezést, hogy a másodfokú tényezőkre bontandó polinom együtthatói valósak. Praktikusan mindenesetre valós együtthatós polinomokra használható az eljárás, mert valós úton számíthatók ki vele a valós együtthatós polinomok konjugált komplex gyökpárjaihoz tartozó valós másodfokú tényezők. Komplex aritmetika alkalmazásával azonban semmi akadálya sincs annak, hogy alkalmas kezdő közelítések ismeretében komplex együtthatós polinomok komplex másodfokú tényezőit kiszámítsuk vele. A komplex eset konvergenciaviszonyai a valóséival megegyeznek. A másik megállapítást a következő szakaszban vizsgáljuk.

2. Konvergenciaviszonyok valós gyökök esetén

Az alábbiakban bemutatjuk a *Bairstow-módszer* levezetésének néhány lépését, hogy a konvergencia vizsgálata során hivatkozhassunk rájuk. A tényezőkre bontandó $a(x)$ polinomot valós együtthatósnak tételezzük fel. Az $a(x)$ polinomot valamely x^2+px+q közelítő másodfokú tényezőjével elosztva fennáll az alábbi azonosság:

$$a(x) \equiv (x^2+px+q)\,b(p,q,x)+c(p,q)x+d(p,q), \tag{2.1}$$

ahol $b(p,q,x)$ x polinomja, az együtthatók p és q függvényei, $c(p,q)$ és $d(p,q)$ pedig az elsőfokú osztási maradék együtthatói. Deriválva (2.1)-et rendre p és q

szerint, az alábbiakat kapjuk:

$$0 \equiv xb(p, q, x)+(x^2+px+q)\frac{\partial b(p, q, x)}{\partial p}+\frac{\partial c(p, q)}{\partial p}x+\frac{\partial d(p, q)}{\partial p}, \tag{2.2}$$

$$0 \equiv b(p, q, x)+(x^2+px+q)\frac{\partial b(p, q, x)}{\partial q}+\frac{\partial c(p, q)}{\partial q}x+\frac{\partial d(p, q)}{\partial q}. \tag{2.3}$$

A (2.3)-ból adódik

$$0 \equiv xb(p, q, x)+(x^2+px+q)\cdot\left(x\frac{\partial b(p, q, x)}{\partial q}+\frac{\partial c(p, q)}{\partial q}\right)- \tag{2.4}$$

$$-\left(\frac{\partial d(p, q)}{\partial q}-p\frac{\partial c(p, q)}{\partial q}\right)x-q\frac{\partial c(p, q)}{\partial q}$$

és minthogy (2.2)-ben és (2.4)-ben a két utolsó tag mint $-xb(p, q, x)$-nek x^2+px+q-val való osztásakor keletkező maradék azonos, azért

$$\frac{\partial c(p, q)}{\partial p}=\frac{\partial d(p, q)}{\partial q}-p\frac{\partial c(p, q)}{\partial q} \quad \text{és} \quad \frac{\partial d(p, q)}{\partial p}=-q\frac{\partial c(p, q)}{\partial q}. \tag{2.5}$$

A p és q szerinti parciális deriváltak ismeretében felírhatjuk a

$$J(p, q)=\frac{\partial(c, d)}{\partial(p, q)}.$$

Jacobi-féle mátrix determinánsát:

$$D(p, q)=q\left(\frac{\partial c(p, q)}{\partial q}\right)^2+\left(\frac{\partial d(p, q)}{\partial q}\right)^2-p\frac{\partial c(p, q)}{\partial q}\frac{\partial d(p, q)}{\partial q}. \tag{2.6}$$

Tényezőkre bontva

$$D(p, q)=\prod_{k=1}^{2}\left(\frac{\partial c(p, q)}{\partial q}x_k+\frac{\partial d(p, q)}{\partial q}\right), \tag{2.7}$$

ahol x_1 és x_2 az x^2+px+q tényező gyökei. A *Bairstow-módszer* az x^2+px+q tényező kiszámítására a $c(p, q)=0$, $d(p, q)=0$ nemlineáris egyenletrendszert oldja meg a *Newton—Raphson-módszer* segítségével. Ismeretes, hogy az eljárás sikeres befejezése esetén, ha p^* és q^* a pontos másodfokú tényező együtthatói, a konvergencia másodrendű, ha $D(p^*, q^*)\neq 0$, és elsőrendű, ha $D(p^*, q^*)=0$. Ez utóbbi esetben

$$\left.\frac{\partial c(p, q)}{\partial q}x_k+\frac{\partial d(p, q)}{\partial q}\right|_{p=p^*, q=q^*}=0 \tag{2.8}$$

x_1^*-ra vagy x_2^*-ra ($x_{1,2}^*$ az $x^2+p^*x+q^*$ másodfokú tényező 2 gyöke). Behelyettesítve p^*-ot, q^*-ot és a megfelelő indexű x_k^*-ot (2.3)-ba, kapjuk a

$$b(p^*, q^*, x_k^*)=0 \tag{2.9}$$

egyenlőséget. Tehát a konvergencia elsőrendű akkor, ha a kiszámított másodfokú tényezőnek és a hányadospolinomnak van közös gyöke.

Ha egy valós együtthatós polinomnak kétszeres és egyszeres valós gyökei vannak, akkor a *Bairstow-eljáráshoz* használt kezdő közelítő értékektől függ, hogy egy kétszeres gyököt tartalmazó másodfokú tényezőhöz első vagy másodrendben konvergál az eljárás. Tekintsük pl. az $a(x)=(x+2)^2(x+3)$ polinomot. Alkalmas iterációs kezdő közelítésekből kiindulva az

$(x+2)^2 = x^2+4x+4$ tényezőhöz másodrendben, az

$(x+2)(x+3) = x^2+5x+6$ tényezőhöz pedig elsőrendben konvergál a Bairstow-eljárás.

Ha történetesen a polinom kétszeres gyökének létezéséről előzetes információ és megfelelő kezdő közelítés áll rendelkezésünkre, akkor a *Bairstow-eljárás* alkalmazása felesleges, mert a kétszeres gyökkel rendelkező másodfokú tényező p és q együtthatói nem függetlenek egymástól ($q=p^2/4$), s az $(x+p/2)^2$ tényező kiszámítására és leválasztására egyszerűbb, másodrendben konvergáló eljárás is létezik (l. [2]), amely kétismeretlenes nemlineáris egyenletrendszer helyett egyetlen nemlineáris egyenlet megoldásával számítja ki a polinom másodfokú tényezőjét.

IRODALOM

[1] BOYD, D. W., "Nonconvergence in Bairstow's method", *SIAM J. Numer. Anal.* **14** (1977).
[2] VARGA, GY., „Kétszeres valós gyökökkel rendelkező valós együtthatós polinomok faktorizálása", *Alk. Mat. Lapok* 4 (1978).

(Beérkezett: 1980. szeptember 23.)

VARGA GYULA
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI
KUTATÓ INTÉZET
1250 BUDAPEST, URI U. 49.

REMARKS ON THE CONVERGENCE OF THE BAIRSTOW METHOD

GY. VARGA

The paper deals with the applicability and the convergence of the Biorstow-method with especial regard to the polynomials which have double real roots.

Alkalmazott Matematikai Lapok 7 (1981) 185—189

EGY SZIVÁRGÁSI FELADAT MEGOLDÁSA

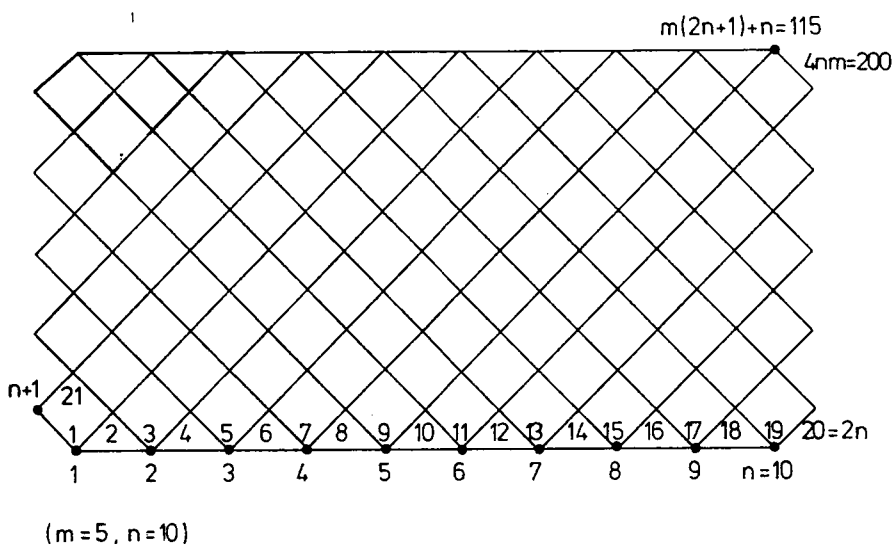
GERGELY JÓZSEF

Budapest

A dolgozatban a felszín alatti víz szivárgásának, valamint a szennyeződés terjedésének modellezésével kapcsolatos matematikai feladat numerikus megoldásával foglalkozunk. A modell alapján nagyméretű ritka mátrixú lineáris egyenletrendszert írunk fel és oldjuk meg azt számítógéppel. A megoldás segítséget ad a szennyeződés terjedésében, sőt magának a porózus közegen keresztül létrejövő vízszállításban mutatkozó véletlen jellegű jelenségek leírásához.

1. Szivárgási feladat modellezése

Szemcsés talajban szivárgó víz mozgásának szimulálására KOVÁCS GYÖRGY akadémikus dolgozott ki fizikai modellt (lásd [1]). A fizikai modell a következőképpen fogalmazható meg: Tekintsük a vizsgált szemcsés talaj egy téglatest darabját. A téglatest egyik függőleges oldalával párhuzamos síkmetszetében a talaj pórusaiból alkotott csatornák síkban elhelyezkedő csőhálózattal modellezhetők (lásd 1. ábra). A talajban a víz áramlását a csőhálózatban való áramlással modellezzük.



1. ábra

Az áramlás iránya lentről felfelé mutat. A csomópontok az áramlás irányára merőleges $2m+1$ egyenes mentén helyezkednek el, míg az alsó és felső egyenesen n számú csomópont van. Minden belső csomópontban négy cső találkozik, kettő szállítja, kettő elvezeti a vizet. A víz mozgása az alsó és felső szint közti nyomáskülönbség hatására jön létre.

Jelölések:

h_i az i-edik csomópontban a nyomás értéke,

d_j a j-edik cső átmérője,

$f_j = a d_j^2$ a j-edik cső keresztmetszeti területe,

$c_j = \frac{b}{f_j^2} = \frac{b}{a^2}\frac{1}{d_j^4}$ a j-edik cső ellenállásának jellemzője.

q_j az áramlás hozama a j-edik csőben

1. Feladat: Az alsó és a felső szinten adott nyomás és adott d_j értékek esetében meghatározandó a nyomás és az áramlás eloszlása, vagyis keresendő a nyomás az összes csomópontban és a hozam az összes csőben.

2. Az egyenletrendszer felállítása

A feladat megoldásához nagyméretű lineáris egyenletrendszer írható fel. Számozzuk be a csomópontokat és a csöveket az 1. ábrán látható módon. Az ábrán az $m=5$ és $n=10$ esetet rajzoltuk fel. Minden belső csomópontban fennáll, hogy az oda érkező és az onnan elfolyó vízhozam összege 0. Vagyis

$$\sum_j s_j q_j = 0 \tag{2.1}$$

minden olyan j-re, amilyen sorszámú cső csatlakozók a vizsgált csomóponthoz, ahol $s_j=1$, ha a j-edik cső a csomópont alatt van és $s_j=-1$, ha felette van. $(m-1)(2n+1)+n+1$ belső csomópontra felírható a (2.1) típusú egyenlet. $4mn$ darab csőre felírható, hogy

$$c_j q_j = h_{\text{kezdő}} - h_{\text{végső}} \tag{2.2}$$

($h_{\text{kezdő}}$ a nyomás értéke a cső kezdő pontjában, $h_{\text{végső}}$ a nyomás értéke a cső végződésénél). Vagyis a csövek végpontjaiban levő nyomások különbsége egyenesen arányos a csőben kialakuló áramlás hozamával és fordítva arányos a keresztmetszeti felület négyzetével.

A (2.1) és a (2.2) összefüggések a következő lineáris egyenletrendszerben fogalmazhatók meg

$$\begin{bmatrix} \mathbf{C} & \mathbf{L} \\ \mathbf{K} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \mathbf{Q} \\ \mathbf{H} \end{bmatrix} = \begin{bmatrix} \mathbf{B1} \\ \mathbf{O} \\ \mathbf{B2} \\ \mathbf{O} \end{bmatrix}. \tag{2.3}$$

$\mathbf{C}$ diagonális mátrix: $\mathbf{C}=\{c_{ij}\}$, $i=1, \ldots, 4mn$. A csőrendszer csatlakozási összefüggéseit a $\mathbf{K}$ és az $\mathbf{L}$ mátrix írja le. Ha

$$\mathbf{K} = \{k_{ij},\ i = 1, \ldots, (m-1)(2n+1)+n+1,\ j = 1, \ldots, 4mn\}$$

és

$$\mathbf{L} = \{l_{ij},\ i = 1, \ldots, 4mn,\ j = 1, \ldots, (m-1)(2n+1)+n+1\},$$

akkor

$$k_{ij} = \begin{cases} 1, & \text{ha az } i\text{-edik csomópontba érkezik a } j\text{-edik cső;} \\ -1, & \text{ha az } i\text{-edik csomópontból kiindul a } j\text{-edik cső;} \\ 0, & \text{különben} \end{cases}$$

és

$$l_{ij} = \begin{cases} 1, & \text{ha az } i\text{-edik cső a } j\text{-edik csomópontból indul;} \\ -1, & \text{ha az } i\text{-edik cső a } j\text{-edik csomópontba fut be;} \\ 0, & \text{különben.} \end{cases}$$

$\mathbf{Q}=\{q_i\}$, $i=1, \ldots, 4mn$;

$\mathbf{H}=\{h_i\}$, $i=1, \ldots, (m-1)(2n+1)+n+1$.

B1 és **B2** alsó és a felső szinten adott nyomások vektora.

3. Az egyenletrendszer megoldása

A (2.3) egyenletrendszer mátrixa ritka. A megoldást ennek felhasználásával végeztük el a következőképpen:

1. lépés: A $\mathbf{C}$ és $\mathbf{L}$ mátrixok segítségével eliminálják a $\mathbf{K}$ mátrix nem zérus elemeit. Ezáltal a kezdetben zéró elemekből álló $\mathbf{0}$ mátrix helyén egy szalagmátrix jelenik meg, aminek $(m-1)(2n+1)+n+1$ sora van, a szalagszélesség pedig $2n+1$.

2. lépés: Megoldjuk a szalagmátrixos lineáris egyenletrendszert. A megoldásra a [2]-ben ismertetett 3/b programot használtuk. A megoldásként a nyomásokat kapjuk.

3. lépés: A 2-ben kiszámolt $\mathbf{H}$ segítségével visszahelyettesítés után kapjuk a $\mathbf{Q}$ vektort, vagyis az áramlás hozamának értékét.

Ezáltal az 1. feladatot megoldottuk.

4. A szennyezettség terjedése

A talajban a vízzel együtt a talajba jutó és vízben oldott vagy keveredett szennyeződés is áramlik. További vizsgálataink célja annak kimutatása, hogy az 1. feladat megoldásából nyert áramlási kép felhasználásával miképpen modellezhetjük a szennyeződés terjedését és a modell alapján milyen áramlási képet számolhatunk a szennyeződés terjedésére. Ehhez vizsgáljuk a következő feladat megoldását:

2. feladat. Az alsó szint egy belépési pontján folyamatosan egységnyi koncentrációjú szennyeződést juttattunk a csőrendszerbe. Hogyan alakul a szennyeződés eloszlása a rendszerben a szennyeződés terjedése szempontjából permanens állapot kialakulása után?

Az egy ponton beáramló szennyeződés szétterjed a csőrendszerben. A terjedés szabályai a következőképpen fogalmazhatók meg:

1. A csomópontokba egy, kettő vagy 3 cső szállíthat vizet. A csomópontban a szennyezettség koncentrációja a lefolyó csöveket jellemző koncentrációértékeknek a hozamok arányában súlyozott átlaga.

2. A csomópontokból egy, kettő vagy három csövön folyhat ki víz. A kifolyó csövek az áramlásaik nagyságainak arányában viszik magukkal a csomópont szennyezettségét, mert a koncentráció minden elfolyó csőben azonos a csomópontra jellemző koncentrációval.

Ezen két szabályt lépésről lépésre alkalmazva az egymásutáni csomópontokra kapjuk a permanens állapotú szennyezés síkbeli eloszlását, majd a legfelső szintnél a kilépő szennyezettség eloszlását.

5. A feladatok számítása

A kitűzött feladatokat a CDC 3300-as gépén oldottuk meg. A program FORTRAN nyelven készült. Az $m=15$ és az $n=30$-as értékek mellett végeztük a számítást, ami 1800 csövet és 945 csomópontot jelent. Az alsó szinten két egységnyi, a felső szinten egységnyi nyomásértéket vettünk fel. A csövek keresztmetszeti felületének négyzetét két különböző (exponenciális ill. közelítőleg egyenletes) eloszlás szerint 0,1-től 10,1-ig 0,2-es lépésközzel választottuk meg. A meghatározott eloszlású véletlen számok segítségével a lehetséges csőméretekből kiválasztottunk $4mn=1800$ darab csövet. Ez alkotta a számításban felhasználható csőkészletet. A csőkészletből minden számításhoz véletlenszerű elrendezéssel alakítottuk ki a számításban résztvevő csőelrendezést.

Az 1. feladat megoldását mindkét eloszlás esetében 15 véletlen csőelrendezés figyelembe vételével számoltuk ki. Ezáltal mindkét vizsgált feladatban (amelyek a feltételezett eloszlásban különböznek egymástól) 15 különböző nyomás és áramlás eloszlást kaptunk. Minden nyomás és áramlás eloszlásból három különböző kiindulású szennyezettség feltevésével számítottuk ki a szennyezettség csőrendszerben való eloszlását. Ezáltal mindkét eloszlás esetén 45 különböző véletlen reprezentációt kaptunk a szennyezettség terjedésére.

Az 1. és 2. feladat megoldásához választott véletlen csőelrendezések esetén megkaptuk az áramlási és nyomás eloszlásokat. Az eredmények statisztikai értékelése lehetőséget ad az [1] dolgozatban vizsgált fizikai modell továbbfejlesztésére.

IRODALOM

[1] KOVÁCS, GY., „A mechanikai diszperzió szerepe a felszín alatti szennyeződés terjedésében", *Hidrológiai szemle* (1980).
[2] GERGELY, J., „Módszerek és programok ritka mátrixokra", *Alkalmazott Matematikai Lapok* **6** (1980) 407—442.

(Beérkezett: 1980. június 16.)

GERGELY JÓZSEF
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI
KUTATÓ INTÉZET
1250 BUDAPEST, URI U. 49.

SOLUTION OF A POLLUTION PROBLEM

J. GERGELY

The mathematical modelling of the influence due to the random character of the structure of equifers is recently investigated with special regard to the distribution of the concentration of pollutants under steady condition and originating from a point source. The numerical solution of the model requires to satisfy a large-scale system of linear equations and the boundary conditions simultaneously. The mathematical handling of this problem is explained in the paper. The solution of the problem by this method enabled the researchers to characterise the random variation of both the hydraulic parameters and the distribution of pollutant depending on the structural variability of aquifers.

A kiadásért felel az Akadémiai Kiadó igazgatója
Műszaki szerkesztő: Sándor István
A kézirat nyomdába érkezett: 1981. IX. 28. — Terjedelem: 16,8 (A/5) ív
81-4125 — Szegedi Nyomda — F. v.: Dobó József igazgató

UTMUTATÁS 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: 26498
ISSN 0133—3399

TARTALOMJEGYZÉK

INDEX

Alkalmazott matematikai lapok

1981/3-4

A MAGYAR TUDOMÁNYOS AKADÉMIA MATEMATIKAI ÉS FIZIKAI TUDOMÁNYOK OSZTÁLYÁNAK KÖZLEMÉNYEI

7.

KÖTET

AKADÉMIAI KIADÓ, BUDAPEST

ALKALMAZOTT MATEMATIKAI LAPOK

A MAGYAR TUDOMÁNYOS AKADÉMIA MATEMATIKAI ÉS FIZIKAI TUDOMÁNYOK OSZTÁLYÁNAK KÖZLEMÉNYEI

VII. 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 7 (1981) 191—204

PRÉKOPA MEGBÍZHATÓSÁGI KÉSZLETMODELLJÉNEK KÉT ÁLTALÁNOSÍTÁSA

KELLE PÉTER

Budapest

Az első olyan készletmodellt PRÉKOPA A. közölte [14], melyben a megrendelt anyag beérkezése véletlen jellegű időpontokban és tételnagyságokban történik. Egyszerű közelítő formulát adott az induló készletszint tervezésére, mellyel mint biztonsági készlettel a folyamatos termelés anyagellátása a rendelési periódusban előírt valószínűséggel biztosítható. A gyakorlatban széles körben alkalmazott modell kétféle általánosítását közöljük.

Elsőként a beérkező véletlen tételnagyságok eloszlásaként tetszőleges eloszlást megengedünk, így egy számítógépes készletnyilvántartási rendszer szokásos adataival a beérkezési folyamat pontosabban becsülhető. Meghatározzuk a folyamatos anyagellátás valószínűségét, mellyel egyúttal az eredeti modell pontos megoldása is megadható. Ezután a modellt általánosítjuk arra az esetre, amikor az anyagfelhasználás pontosan nem tervezhető és azt valószínűségi változónak tekintjük. Normális eloszlású becslési hiba esetére egyszerű közelítő formulát adunk a szükséges induló készletszintre.

A fenti modellek egy számítógépes készletgazdálkodási döntéselőkészítő programrendszerbe épülnek, mely az MTA Számítástechnikai és Automatizálási Kutató Intézetének Operációkutatási Osztályán készült és a Dunai Vasműben már részben bevezetésre került.

1. Bevezetés

A készletmodellek alkalmazásának igénye és felhasználási területe az utóbbi években hazánkban is megnőtt. Ez szükségessé teszi a korábban már jól bevált modellek általánosítását oly módon, hogy a gyakorlati alkalmazási lehetőségeik is bővüljenek.

Olyan készletmodellekkel foglalkozunk, melyeknél a rendelés feladása után a megrendelt anyag (termék, áru) beérkezése nem egy tételben történik, hanem a rendelési periódus véletlen jellegű időpontjaiban. Az első ilyen modellt, melyben a beérkező tételnagyságok is véletlen jellegűek PRÉKOPA A. [14] közölte. Az egyszerű formulával megadott közelítő megoldás a gyakorlatban igen széles körben elterjedt. A modell kétféle, a gyakorlatban szintén jól felhasználható általánosítását írjuk le cikkünkben, mellyel egyúttal az eredeti modell pontos megoldását is meg tudjuk adni. Először a gyakorlati feladatot és a modell kialakulásának előzményeit ismertetjük.

A vizsgált vállalat folyamatos termelésének anyagszükségletét rögzített T időközönként (pl. negyedévenként) történő rendelésfeladással biztosítja. Az ellátó vállalat a megrendelt C mennyiség beszállítását a periódus végére vállalja az előszállítási jog fenntartása mellett. A gyakorlatban általában ez azt jelenti, hogy a rendelési periódus (pl. negyedév) folyamán több alkalommal van anyagbeérkezés, melynek időpontjai és tételnagyságai a rendelésfeladáskor előre nem ismertek. A vállalat egy M nagyságú záró készletet tervez, mely a következő periódusban mint induló készlet biztosítja a termelés zavartalan anyagellátását. A feladat ennek a biztonsági kész-

letül szolgáló induló készletszintnek a meghatározása. A költségtényezők, elsősorban a hiánykőltség meghatározásának nehézsége miatt megbízhatósági kritériumot kell alkalmazni: a vállalat olyan minimális induló készletszintet kíván tervezni, mely a folyamatos anyagellátást előírt valószínűséggel biztosítja.

A fenti gyakorlati feladat első modelljét PRÉKOPA A. [14] és ZIERMANN M. [23] közölte. Feltételezésük szerint a rendelés beérkezése a $[0, T]$ rendelési periódus bármelyik időpontjában egyenlő valószínűséggel történhet. A beérkezések száma ismert, ezt n jelöli. Az egymást követő véletlen beérkezési időpontok modellje így a $[0, T]$ intervallumon egyenletes eloszlásból vett n számú véletlen pont nagyság szerint rendezett sorozata: $t_1^* < t_2^* < \ldots < t_n^*$ az ún. rendezett minta. A beérkező tételnagyságokat egyenlőknek tételezik fel, az anyagfelhasználás intenzitása pedig állandó és ismert a $[0, T]$ intervallumon.

Az egységek alkalmas megválasztásával $T=1$ és $C=1$ teljesül, így az általánosság megszorítása nélkül a továbbiakban is ezzel az egyszerűsítő jelöléssel élünk. A beérkezési folyamat fenti modellje így a $[0, 1]$ intervallumon egyenletes eloszlás empirikus eloszlásfüggvénye, melynek szokásos jelölése $F_n(t)$.

Dolgozatunk PRÉKOPA A. [14] modelljéhez kapcsolódik, mely a fentitől abban különbözik, hogy a beérkező tételnagyságok is véletlen jellegűek. Van egy $0 \leqq \delta \leqq 1/n$ minimális mennyiség, mely minden szállítás alkalmával biztosan beérkezik. A fennmaradó $1-n\delta$ mennyiség a beszállítási időpontok között véletlenszerűen oszlik meg. A modell szerint ez a felosztás $n-1$ számú, a $[0, 1-n\delta]$ intervallumon egyenletes eloszlású véletlen ponttal történik. Ezek a véletlen mennyiségek hozzáadódnak a fix δ mennyiséghez, így kapjuk a beérkezési tételnagyságokat. A beérkezési folyamat PRÉKOPA A. [14] szerinti modellje így

$$F_n(t, \lambda) = \begin{cases} 0, & \text{ha } 0 \leqq t \leqq t_1^*, \\ \lambda \dfrac{k}{n} + (1-\lambda)\tau_k^*, & \text{ha } t_k^* < t \leqq t_{k+1}^*, \\ & k = 1, \ldots, n-1, \\ 1, & \text{ha } t_n^* < t \leqq 1, \end{cases} \tag{1.1}$$

ahol $\lambda = n\delta$ a beérkező tételnagyságok determinisztikus hányada és $\tau_1^* < \tau_2^* < \ldots < \tau_{n-1}^*$ pedig a $[0, 1]$ intervallumon egyenletes eloszlásból vett $n-1$ elemű rendezett véletlen minta. A $\lambda = 1$ esetben ez az ismert $F_n(t)$ empirikus eloszlásfüggvényt adja.

A megbízhatósági szintet a $[0, 1]$-gyel jelölt rendelési periódusban a folyamatos anyagellátás valószínűségével mérjük. Azt a legkisebb M induló készletet keressük, melyre a folyamatos anyagellátás valószínűsége az előírt $1-\varepsilon$ valószínűségi szintet eléri, azaz

$$P\left(\sup_{0 \leqq t \leqq 1} \{t - F_n(t, \lambda)\} \leqq M\right) \geqq 1-\varepsilon. \tag{1.2}$$

A közelítő megoldást PRÉKOPA A. [14] cikkében a következő formulával adja meg:

$$M \approx \sqrt{1+(1-\lambda)^2}\sqrt{\frac{1}{2n}\ln\frac{1}{\varepsilon}}, \tag{1.3}$$

melynek egyszerűsége és költségtényezőkből való függetlensége széles körű gyakorlati felhasználását elősegítette. A közelítő megoldás SZMIRNOV [19] ismert határ-

eloszlás tételének a következő általánosításán alapul $(0<y<1)$:

$$\lim_{n\to\infty} P\left(\sqrt{\frac{n}{1+(1-\lambda)^2}}\sup_{0\leqq t\leqq 1}\{t-F_n(t,\lambda)\}\leqq y\right)=1-e^{-2y^2}, \tag{1.4}$$

melyet PRÉKOPA A. [16] cikkében bizonyít. Az (1.3) közelítő megoldás pontossága kis n $(n<10)$ értékekre nem kielégítő.

Z. V. BIRNBAUM és H. F. TINGEY [1] adja meg a

$$\sup_{0\leqq t\leqq 1}\{t-F_n(t,\lambda)\} \tag{1.5}$$

valószínűségi változó pontos eloszlását a $\lambda=1$ esetre (az $F_n(t)$ empirikus eloszlásfüggvényre), LÁSZLÓ Z. [8] pedig a $\lambda=0$ esetre (amikor bármilyen kicsi tételnagyság előfordulhat). Az (1.5) valószínűségi változó eloszlása tetszőleges $0\leqq\lambda\leqq 1$ esetén először LÁSZLÓ Z. [9] nem publikált előadásában szerepelt. Cikkünkben ez egy általánosabb modell megoldásának speciális eseteként adódik.

A fenti gyakorlati feladat megoldásához kapcsolódik még többek között GERENCSÉR L. [3], KELLE P. [5, 6], MÓRITZ A. [10], NÉMETH GY. [11], PINTÉR J. [12, 13] és VASS I. [21] cikke, és a témakör elméleti és gyakorlati eredményeit összefoglaló KELLE P. [7] tanulmány.

2. A véletlen beérkezési folyamat általánosítása

A vállalati számítógépes készletnyilvántartási rendszerek rögzítik és archiválják a raktárba érkező tételek nagyságát. A megőrzött információt fel lehet használni a következő periódusban történő véletlen nagyságú szállítási tételek eloszlásának becslésére és így a beérkezési folyamat pontosabb leírására. Ehhez azonban a beérkezési folyamat PRÉKOPA A. által adott (1.1) modeljének általánosítása szükséges.

A beérkező tételnagyságok eloszlása az általánosított modellben tetszőleges lehet. Az i-edik szállítási időpontban beérkező mennyiséget β_i/n jelöli, így a szállítási tételnagyságokat a $\boldsymbol{\beta}'=(\beta_1,\ldots,\beta_n)$ véletlen vektor jellemzi. A beérkezési folyamat általánosított modellje (a $t^*_{n+1}=1$ jelöléssel)

$$F_n^{\boldsymbol{\beta}}(t)=\begin{cases}0, & \text{ha}\quad 0\leqq t\leqq t_1^*,\\ \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\beta_i, & \text{ha}\quad t_k^*<t\leqq t_{k+1}^*,\\ & k=1,\ldots,n.\end{cases} \tag{2.1}$$

Itt is feltételezzük, mint a korábbi (1.1) modellnél, hogy a beérkezések $t_1^*<t_2^*<\ldots<t_n^*$ időpontjai úgy tekinthetők, mint a $[0,1]$ beérkezési időintervallumon egyenletes eloszlásból származó n elemű véletlen minta nagyság szerint rendezett elemei. Ez a gyakorlati feladatok nagy részében megengedhető közelítés. A (2.1) kifejezés szerint definiált $F_n^{\boldsymbol{\beta}}(t)$ sztochasztikus folyamat az egyenletes eloszlásból származó empirikus eloszlásfüggvény általánosított alakja, melyben a lépcsős függvény ugrásai véletlen β_i/n nagyságúak a szokásos $1/n$ nagyságú ugrások helyett. Ezt a folyamatot először CSÁKI E. [2] vizsgálta és véletlen ugrású empirikus eloszlásfüggvénynek nevezte.

Amennyiben a beérkezési időpontok eloszlása nem tekinthető egyenletesnek a beérkezési időintervallumon, a fenti modell transzformálható arra az esetre, melyben a β_i/n $(i=2, \ldots, n)$ tetszőleges eloszlású véletlen számok az egymást követő beérkezési időpontok közötti időtartamokat képviselik (illetve β_1/n a periódus kezdetétől az első beérkezésig terjedő időtartamot), melyek eloszlása a számítógépes készletnyilvántartási rendszer archivált raktári beérkezési időpontjai alapján becsülhető. Ez esetben a megrendelt mennyiség (mely korábbi megállapításunk szerint egységnyinek tekinthető) beérkezésének mennyiségi megoszlása a beérkezési időpontok között a [0, 1] intervallum $n-1$ számú véletlen, egyenletes eloszlású ponttal történő felosztásának felel meg.

Abban az esetben, amikor a véletlen beérkezési időpontok és a véletlen tételnagyságok egyike sem tekinthető egyenletes eloszlásúnak, sokkal bonyolultabb modell konstrukcióra és megoldási módszerre van szükség. Egy ilyen modellt közöl PRÉKOPA A. [15], melynek *Monte Carlo-módszert* is felhasználó megoldása PRÉKOPA A.—KELLE P. [17] cikkében szerepel.

3. A folyamatos anyagellátás valószínűségének meghatározása

M nagyságú induló készlet esetén a [0, 1] rendelési periódus alatt folyamatos az anyagellátás, ha a $\xi(t) \leqq \eta(t)+M$ egyenlőtlenség teljesül minden $0 \leqq t \leqq 1$ értékre, azaz

$$\sup_{0 \leqq t \leqq 1} \{\xi(t)-\eta(t)\} \leqq M, \tag{3.1}$$

ahol $\eta(t)$ jelöli a beérkező, $\xi(t)$ pedig az igényelt mennyiséget a $[0, t]$ intervallumban, $0 \leqq t \leqq 1$.

Vizsgáljuk meg azt az esetet, amikor az $\eta(t)$ beérkezési folyamat a (2.1) alakban kifejezhető, az igényfolyamat pedig a folyamatos termelés konstans α $(\alpha>0)$ intenzitású anyagfelhasználása, melyet először ismertnek tételezünk fel. Tehát

$$\xi(t) = \alpha t, \quad 0 \leqq t \leqq 1, \tag{3.2}$$

ahol α nem szükségképpen 1, mint a korábbi modelleknél, ahol a vizsgált [0, 1] intervallumban beérkező és felhasznált mennyiségnek meg kellett egyezni. Ennek különös jelentősége akkor van, ha a termelés intenzitása a rendelés időpontjában pontosan nem becsülhető (l. következő pontban).

A beérkezési és az igényfolyamat egy lehetséges realizációját és az ennek megfelelő készletalakulást az 1. ábrán szemléltetjük adott M induló készletszint esetén.

A folyamatos anyagellátás valószínűségének meghatározása a

$$\sup_{0 \leqq t \leqq 1} \{\xi(t)-\eta(t)\}$$

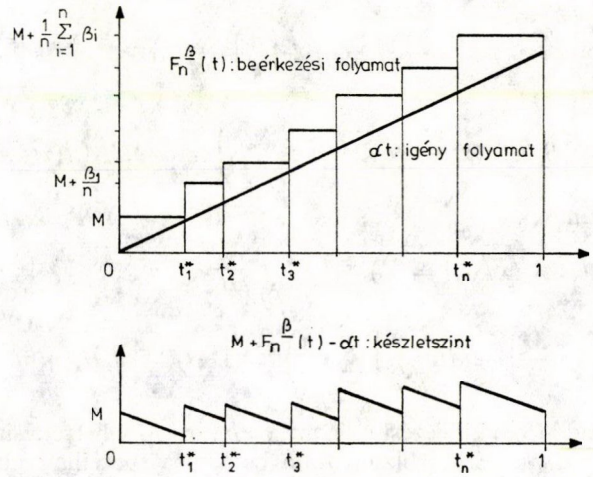
valószínűségi változó eloszlásának megadását jelenti, mely általában igen nehéz feladat. A $\xi(t)=t$ és $\eta(t)=F_n(t)$ esetben az empirikus eloszlásfüggvényre vonatkozó jól ismert *Kolmogorov—Szmirnov-statisztikát* (l. pl. RÉNYI [18]) kapjuk, melynek egy modellünkre általánosított változatát a következő formában tudjuk igazolni:

3.1. TÉTEL: Ha a β_i $(i=1, \ldots, n)$ valószínűségi változók felcserélhető növekményűek, azaz $n!$ számú permutációik mindegyikének azonos az együttes eloszlása, akkor $M>0$ esetén

$$P\Big(\sup_{0\leqq t\leqq s}\{\alpha t - F_n(t)\} \leqq M\Big) = 1-\Big(1-\frac{M}{\alpha}\Big)^n - \tag{3.3}$$

$$-\frac{M}{\alpha}\sum_{k=1}^{n}\binom{n}{k}\int_0^{n(\alpha s-M)}\left(\frac{M+\frac{x}{n}}{\alpha}\right)^{k-1}\left(1-\frac{M+\frac{x}{n}}{\alpha}\right)^{n-k} dH_k(x),$$

ahol $H_k(x)$ jelöli a $\sum_{i=1}^{k}\beta_i$ valószínűségi változó eloszlásfüggvényét $(k=1, \ldots, n)$, $s\leqq 1$, $\alpha>0$ és $F_n^{\beta}(t)$ a (2.1) szerint definiált véletlen ugrású empirikus eloszlásfüggvény.



1. ábra

Bizonyítás: A bizonyítás L. TAKÁCS [20, 17. fejezet, 1. tétel] következő tételén alapul:

Ha $\chi(t)$, $0\leqq t\leqq T$ szeparábilis, felcserélhető növekményű sztochasztikus folyamat, melynek majdnem minden realizációja nemcsökkenő lépcsős függvény és a $t=0$ pontban értéke nulla, akkor $0<y\leqq T_1\leqq T$ esetén

$$P\Big(\sup_{0\leqq t\leqq T_1}\{t-\chi(t)\} \leqq y\Big) = 1-\int_y^{T_1}\frac{y}{t}\,d_tP(\chi(t)\leqq t-y), \tag{3.4}$$

ahol

$$d_tP(\chi(u)\leqq t) = P(t<\chi(u)<t+\alpha t)$$

függetlenül attól, hogy u függ-e t-től.

A $\chi(t)=F_n^{\beta}(t)/\alpha$ ($\alpha>0$ konstans) sztochasztikus folyamatra L. TAKÁCS fenti tételének mindegyik feltétele teljesül, amennyiben a β_i ($i=1, \ldots, n$) valószínűségi változók felcserélhető növekményűek. Az $y=M/\alpha$ helyettesítéssel

$$d_t P(\chi(t) \leqq t-y) = d_t P(F_n^{\beta}(t) \leqq \alpha t - M).$$

Az $\alpha t-M$ egyenes ($0 \leqq t \leqq s$) a 0 magasságban (az első ugrás előtt) metszi az $F_n^{\beta}(t)$ lépcsős függvényt, ha $t_1^*>M/\alpha$, mely eseménynek $(1-M/\alpha)^n$ a valószínűsége $t \geqq M/\alpha$ esetén és 0 egyébként (l. 1. ábrán!). Így

$$d_t P(F_n^{\beta}(t) \leqq \alpha t - M \,|\, t = M/\alpha < t_1^*) = (1-M/\alpha)^n.$$

A metszés a k-adik ugrás magasságában történik ($k=1, \ldots, n$), ha $t_k^*<t \leqq t_{k+1}^*$ (ahol $t_{n+1}^*=1$ a definíció szerint) és

$$\sum_{i=1}^{k} \beta_i/n \leqq \alpha t - M,$$

melyre

$$P(t_k^* < t \leqq t_{k+1}^*) = \binom{n}{k} t^k (1-t)^{n-k}$$

és

$$d_t P\Big(\sum_{i=1}^{k} \beta_i \leqq n(\alpha t - M) \,\Big|\, t_k^* < t \leqq t_{k+1}^* \Big) = n\alpha\, d_t H_k(n(\alpha t - M)).$$

Végül (3.4) alapján

$$P\Big(\sup_{0 \leqq t \leqq s} \{\alpha t - F_n^{\beta}(t)\} \leqq M\Big) = 1 - \int_{M/\alpha}^{s} \frac{M}{\alpha t}\, d_t P(F_n^{\beta}(t) \leqq \alpha t - M) =$$

$$= 1 - \Big(1-\frac{M}{\alpha}\Big)^n - Mn \int_{M/\alpha}^{s} \Big[\sum_{k=1}^{n} \binom{n}{k} t^{k-1}(1-t)^{n-k}\Big]\, d_t H_k(n(\alpha t - M)),$$

mely a bizonyítandó (3.3) kifejezést adja az $x=n(\alpha t-M)$ helyettesítést alkalmazva.

A fenti tételt CSÁKI E. [2] bizonyította be arra a speciális esetre, amikor a β_i ($i=1, \ldots, n$) valószínűségi változók függetlenek és azonos eloszlásúak.

A tervezési periódus várható anyagigényének, a beszállítások várható számának ismeretében a 3.1. tétel felhasználható az induló készletszint tervezésére, amennyiben a beérkező tételnagyságok eloszlására becslést tudunk adni. A gyakorlatban ezt a becslést általában a számítógépes készletnyilvántartási rendszer archivált beérkezési tételnagyságai alapján lehet elvégezni. Néhány (8—12) múltbeli periódus megfigyelt adataira célszerű az eloszlást illeszteni. Kevés számú beérkezés esetén (2—4) a relatív gyakoriságok jöhetnek szóba.

Az előírt $1-\varepsilon$ megbízhatósági szinthez szükséges M induló készletnagyságot a

$$P\Big(\sup_{0 \leqq t \leqq 1} \{\alpha t - F_n^{\beta}(t)\} \leqq M\Big) = 1-\varepsilon \tag{3.5}$$

egyenlet megoldása adja. Ezt a szokásos numerikus egyenletmegoldó módszerek bármelyikével a gyakorlat számára szükséges pontossággal igen gyorsan elvégezhetjük a 3.1. tétel eredményének felhasználásával.

Az $s \leqq 1$ paraméter megengedi azt a lehetőséget is, hogy a rendelési periódusnak csak egy részén kívánjuk meg a folyamatos anyagellátást. A gyakorlatban azonban az $s=1$ érték jön szóba általában.

A $F_n^{\beta}(t)$ beérkezési folyamat általános modellje szerint az is lehetséges, hogy a megrendelt mennyiség a rendelési periódus szerződésben rögzített végpontjáig sem érkezzék be teljes egészében, mint az a gyakorlatban is előfordul. A megrendelés határidőre történő teljes beszállítása is előírható. Ez esetben természetesen a β_i $(i=1, \ldots, n)$ valószínűségi változók nem függetlenek, de néhány fontos esetben felcserélhető növekményűek. Erre példa a PRÉKOPA A. [14] által adott $F_n(t, \lambda)$ beérkezési folyamat, melyet az (1.1) formában definiál. Az $F_n(t, \lambda)$ folyamat speciális esete a (2.1) szerinti $F_n^{\beta}(t)$ sztochasztikus folyamatnak, melyben a

$$\beta_i = \lambda + n(1-\lambda)(\tau_i^* - \tau_{i-1}^*), \quad i = 1, \ldots, n \tag{3.6}$$

valószínűségi változók szerepelnek (ahol $\tau_0^*=0$, ill. $\tau_n^*=1$ definíció szerint).

A 3.1. tétel alkalmazásával megadható a folyamatos anyagellátás valószínűségének pontos értéke az $F_n(t, \lambda)$ beérkezési és a $\xi(t)=\alpha t$ igényfolyamat esetére:

3.2. TÉTEL. Az (1.1) kifejezés szerint definiált sztochasztikus folyamatra $0 \leqq \lambda \leqq 1$ és $\max\{0, \alpha-1\} < M < \alpha s$ esetén

$$P\left(\sup_{0 \leqq t \leqq s} \{\alpha t - F_n(t, \lambda)\} \leqq M\right) = 1 - \left(1 - \frac{M}{\alpha}\right)^n - \tag{3.7}$$

$$- \frac{M}{\alpha} \sum_{k=1}^{r} k \binom{n}{k} \binom{n-1}{k} \int_0^{a_k} \left(\frac{M + (1-\lambda)z + \lambda \frac{k}{n}}{\alpha} \right)^{k-1} *$$

$$* \left(1 - \frac{M + (1-\lambda)z + \lambda \frac{k}{n}}{\alpha} \right)^{n-k} z^{k-1}(1-z)^{n-k-1} dz,$$

ahol $a_k = \min\left\{ \dfrac{\alpha s - M - \lambda \frac{k}{n}}{1-\lambda}, 1 \right\}$ és $r = \min\left\{ \left[\dfrac{n}{\lambda}(\alpha s - M) \right], n-1 \right\}$.

A $\lambda=0$ esetén $r=n-1$ és $\lambda=1$ esetén $a_k=1$ $(k=1, \ldots, n)$.

Bizonyítás: A (3.6) kifejezés által definiált β_i valószínűségi változók felcserélhető növekményűek, mert a minta blokkoknak nevezett $\tau_i^* - \tau_{i-1}^*$ $(i=1, \ldots, n)$ valószínűségi változók együttes sűrűségfüggvénye $f(\mathbf{x})=1$ a $\mathbf{0} \leqq \mathbf{x} \leqq \mathbf{1}$ n dimenziós kockán és $f(\mathbf{x})=0$ ezen kívül. (L. pl. S. WILKS [22, 8.7 fejezet]). Így a 3.1. tétel alkalmazható.

A β_i $(i=1, \ldots, n)$ valószínűségi változók (3.6) szerinti értelmezése miatt

$$H_k(x) = P\left(\tau_k^* < \frac{x - k\lambda}{n(1-\lambda)} \right)$$

1. TÁBLÁZAT

$1-\varepsilon=$		0,75					0,80					0,85					0,90				
N:	$\lambda=$	0,00	0,25	0,50	0,75	1,00	0,00	0,25	0,50	0,75	1,00	0,00	0,25	0,50	0,75	1,00	0,00	0,25	0,50	0,75	1,00
4	P:	0,470	0,434	0,406	0,387	0,382	0,509	0,469	0,438	0,418	0,412	0,553	0,509	0,476	0,455	0,447	0,606	0,558	0,522	0,501	0,493
	K:	0,589	0,520	0,465	0,429	0,416	0,634	0,561	0,501	0,462	0,449	0,689	0,609	0,544	0,502	0,487	0,759	0,671	0,600	0,553	0,536
	H:	25,1	19,8	14,7	10,9	8,9	24,7	19,5	14,5	10,5	8,7	24,6	19,5	14,4	10,3	8,9	25,2	20,2	14,8	10,3	8,8
5	P:	0,431	0,396	0,368	0,349	0,343	0,466	0,428	0,397	0,378	0,370	0,507	0,465	0,432	0,411	0,404	0,557	0,510	0,475	0,453	0,447
	K:	0,526	0,465	0,416	0,384	0,372	0,567	0,501	0,449	0,414	0,401	0,616	0,544	0,487	0,449	0,436	0,679	0,600	0,536	0,495	0,480
	H:	22,2	17,7	13,2	9,9	8,5	21,6	17,3	12,9	9,5	8,3	21,4	17,2	12,8	9,2	7,9	21,7	17,5	13,0	9,1	7,3
6	P:	0,400	0,366	0,339	0,321	0,315	0,433	0,396	0,366	0,347	0,341	0,472	0,430	0,398	0,378	0,372	0,519	0,473	0,439	0,418	0,410
	K:	0,481	0,425	0,380	0,350	0,340	0,518	0,458	0,409	0,377	0,366	0,562	0,497	0,445	0,410	0,398	0,619	0,548	0,490	0,452	0,438
	H:	20,0	16,1	12,1	9,1	8,1	19,5	15,7	11,8	8,7	7,5	19,2	15,5	11,6	8,4	7,0	19,3	15,7	11,6	8,1	6,7
7	P:	0,376	0,342	0,316	0,299	0,293	0,407	0,370	0,342	0,323	0,317	0,443	0,403	0,372	0,352	0,346	0,488	0,443	0,410	0,389	0,382
	K:	0,445	0,393	0,352	0,324	0,315	0,479	0,424	0,379	0,349	0,339	0,521	0,460	0,412	0,379	0,368	0,573	0,507	0,453	0,418	0,406
	H:	18,3	14,8	11,2	8,5	7,4	17,8	14,4	10,9	8,1	6,8	17,5	14,2	10,6	7,7	6,5	17,5	14,4	10,6	7,5	6,3
8	P:	0,356	0,323	0,298	0,281	0,276	0,385	0,350	0,322	0,304	0,298	0,419	0,380	0,350	0,331	0,324	0,462	0,419	0,386	0,366	0,358
	K:	0,416	0,368	0,329	0,303	0,294	0,448	0,396	0,355	0,327	0,317	0,487	0,430	0,385	0,355	0,344	0,536	0,474	0,424	0,391	0,379
	H:	17,0	13,9	10,5	7,9	6,8	16,5	13,4	10,2	7,5	6,4	16,2	13,2	9,9	7,2	6,1	16,1	13,2	9,8	7,0	5,9
9	P:	0,338	0,307	0,282	0,266	0,261	0,366	0,332	0,305	0,288	0,282	0,399	0,361	0,332	0,313	0,307	0,440	0,398	0,366	0,346	0,339
	K:	0,392	0,347	0,310	0,286	0,278	0,423	0,374	0,334	0,308	0,299	0,459	0,406	0,363	0,335	0,325	0,506	0,447	0,400	0,369	0,358
	H:	16,0	13,0	9,9	7,5	6,4	15,4	12,6	9,6	7,1	6,1	15,1	12,3	9,3	6,8	5,8	14,9	12,3	9,2	6,6	5,5
10	P:	0,324	0,293	0,269	0,253	0,248	0,350	0,317	0,291	0,274	0,268	0,382	0,345	0,317	0,298	0,292	0,421	0,380	0,349	0,329	0,323
	K:	0,372	0,329	0,294	0,271	0,263	0,401	0,355	0,317	0,292	0,284	0,435	0,385	0,344	0,317	0,308	0,480	0,424	0,379	0,350	0,339
	H:	15,1	12,4	9,4	7,1	6,1	14,5	11,9	9,1	6,8	5,8	14,1	11,6	8,8	6,4	5,5	14,0	11,5	8,6	6,2	5,2
11	P:	0,311	0,281	0,258	0,242	0,237	0,336	0,304	0,278	0,262	0,256	0,366	0,331	0,303	0,285	0,279	0,404	0,365	0,334	0,315	0,308
	K:	0,355	0,314	0,281	0,259	0,251	0,382	0,338	0,302	0,279	0,270	0,415	0,367	0,328	0,303	0,294	0,457	0,404	0,362	0,333	0,324
	H:	14,3	11,8	9,0	6,8	5,9	13,8	11,3	8,6	6,5	5,6	13,4	11,0	8,3	6,1	5,2	13,2	10,9	8,2	5,9	4,9
12	P:	0,299	0,270	0,247	0,233	0,227	0,324	0,292	0,268	0,251	0,246	0,353	0,318	0,291	0,274	0,268	0,389	0,351	0,321	0,302	0,296
	K:	0,340	0,300	0,269	0,248	0,240	0,366	0,324	0,290	0,267	0,259	0,398	0,351	0,314	0,290	0,281	0,438	0,387	0,346	0,319	0,310
	H:	13,6	11,2	8,6	6,5	5,6	13,1	10,8	8,2	6,2	5,3	12,7	10,5	7,9	5,9	5,0	12,5	10,3	7,8	5,6	4,7
13	P:	0,289	0,261	0,238	0,224	0,219	0,313	0,282	0,258	0,242	0,237	0,341	0,307	0,281	0,264	0,258	0,376	0,339	0,310	0,291	0,285
	K:	0,327	0,289	0,258	0,238	0,231	0,352	0,311	0,278	0,256	0,249	0,382	0,338	0,302	0,278	0,270	0,421	0,372	0,333	0,307	0,298
	H:	13,0	10,8	8,3	6,3	5,4	12,5	10,3	7,9	5,9	5,1	12,1	10,0	7,6	5,6	4,8	11,9	9,8	7,4	5,4	4,5

14	P:	0,280	0,252	0,230	0,216	0,211	0,303	0,273	0,249	0,234	0,229	0,330	0,297	0,271	0,254	0,249	0,364	0,328	0,299	0,281	0,275
	K:	0,315	0,278	0,249	0,229	0,223	0,339	0,300	0,268	0,247	0,240	0,368	0,325	0,291	0,268	0,260	0,405	0,358	0,321	0,296	0,287
	H:	12,5	10,4	8,0	6,1	5,2	12,0	9,9	7,6	5,7	4,9	11,6	9,6	7,3	5,4	4,6	11,4	9,4	7,1	5,2	4,3
15	P:	0,271	0,244	0,223	0,209	0,205	0,294	0,264	0,241	0,226	0,221	0,320	0,288	0,263	0,246	0,241	0,353	0,318	0,290	0,272	0,266
	K:	0,304	0,269	0,240	0,222	0,215	0,328	0,290	0,259	0,239	0,232	0,356	0,314	0,281	0,259	0,251	0,392	0,346	0,310	0,286	0,277
	H:	12,1	10,0	7,7	5,9	5,0	11,6	9,6	7,3	5,5	4,7	11,2	9,2	7,0	5,2	4,5	10,9	9,0	6,8	5,0	4,2
16	P:	0,264	0,237	0,217	0,203	0,198	0,285	0,257	0,234	0,219	0,214	0,311	0,279	0,255	0,239	0,233	0,343	0,308	0,281	0,264	0,258
	K:	0,294	0,260	0,233	0,215	0,208	0,317	0,280	0,251	0,231	0,224	0,344	0,304	0,272	0,251	0,243	0,379	0,335	0,300	0,277	0,268
	H:	11,6	9,7	7,4	5,7	4,9	11,2	9,3	7,1	5,4	4,6	10,7	8,9	6,8	5,1	4.3	10,5	8,7	6,6	4,8	4,1
17	P:	0,257	0,231	0,211	0,197	0,193	0,278	0,250	0,228	0,213	0,208	0,303	0,272	0,248	0,232	0,227	0,334	0,300	0,274	0,256	0,250
	K:	0,286	0,252	0,226	0,208	0,202	0,308	0,272	0,243	0,224	0,218	0,334	0,295	0,264	0,243	0,236	0,368	0,325	0,291	0,268	0,260
	H:	11,3	9,4	7,2	5,5	4,7	10,8	9,0	6,9	5,2	4,5	10,4	8,6	6,6	4,9	4,2	10,1	8,4	6,4	4,7	3,9
18	P:	0,250	0,225	0,205	0,192	0,188	0,271	0,243	0,222	0,207	0,203	0,295	0,265	0,241	0,226	0,221	0,326	0,292	0,266	0,249	0,244
	K:	0,277	0,245	0,219	0,202	0,196	0,299	0,264	0,236	0,218	0,211	0,325	0,287	0,257	0,237	0,230	0,358	0,316	0,283	0,261	0,253
	H:	10,9	9,1	7,0	5,4	4,6	10,4	8,7	6,7	5,0	4,3	10,0	8,3	6,4	4,8	4,1	9,8	8,1	6,2	4,5	3,6
19	P:	0,244	0,219	0,200	0,187	0,183	0,264	0,237	0,216	0,202	0,197	0,288	0,258	0,235	0,220	0,215	0,318	0,285	0,260	0,243	0,238
	K:	0,270	0,239	0,214	0,197	0,191	0,291	0,257	0,230	0,212	0,206	0,316	0,279	0,250	0,230	0,223	0,348	0,308	0,275	0,254	0,246
	H:	10,6	8,8	6,8	5,2	4,5	10,1	8,4	6,5	4,9	4,2	9,7	8,1	6,2	4,6	4,0	9,4	7,9	6,0	4,4	3,6
20	P:	0,239	0,214	0,195	0,183	0,178	0,258	0,232	0,211	0,197	0,183	0,281	0,252	0,230	0,215	0,210	0,311	0,279	0,254	0,237	0,232
	K:	0,263	0,233	0,208	0,192	0,186	0,284	0,251	0,224	0,207	0,201	0,308	0,272	0,243	0,224	0,218	0,339	0,300	0,268	0,247	0,240
	H:	10,3	8,6	6,6	5,1	4,4	9,9	8,2	6,3	4,8	4,1	9,5	7,9	6,0	4,5	3,9	9,2	7,6	5,8	4,3	3,5
21	P:	0,233	0,210	0,191	0,178	0,174	0,253	0,227	0,206	0,193	0,188	0,275	0,247	0,224	0,210	0,205	0,304	0,272	0,248	0,232	0,226
	K:	0,257	0,227	0,203	0,187	0,182	0,277	0,245	0,219	0,202	0,196	0,301	0,266	0,238	0,219	0,213	0,331	0,293	0,262	0,241	0,234
	H:	10,1	8,4	6,5	5,0	4,3	9,6	8,0	6,2	4,7	4,0	9,2	7,7	5,9	4,4	3,6	8,9	7,4	5,6	4,2	3,4
22	P:	0,229	0,205	0,187	0,174	0,170	0,247	0,222	0,202	0,189	0,184	0,269	0,242	0,220	0,205	0,201	0,298	0,267	0,242	0,227	0,221
	K:	0,251	0,222	0,198	0,183	0,178	0,270	0,239	0,214	0,197	0,191	0,294	0,260	0,232	0,214	0,208	0,323	0,286	0,256	0,236	0,229
	H:	9,8	8,2	6,3	4,9	4,2	9,4	7,8	6,0	4,6	3,9	9,0	7,5	5,7	4,3	3,5	8,7	7,2	5,5	4,0	3,3
23	P:	0,224	0,201	0,183	0,171	0,167	0,242	0,217	0,198	0,185	0,180	0,264	0,237	0,215	0,201	0,196	0,292	0,261	0,237	0,222	0,217
	K:	0,245	0,217	0,194	0,179	0,174	0,264	0,234	0,209	0,193	0,187	0,287	0,254	0,227	0,209	0,203	0,316	0,280	0,250	0,231	0,224
	H:	9,6	8,0	6,2	4,7	4,1	9,1	7,6	5,9	4,5	3,8	8,7	7,3	5,6	4,2	3,4	8,4	7,0	5,4	4,0	3,2
24	P:	0,220	0,197	0,179	0,167	0,163	0,238	0,213	0,194	0,181	0,177	0,259	0,232	0,211	0,197	0,192	0,286	0,256	0,233	0,217	0,212
	K:	0,240	0,212	0,190	0,175	0,170	0,259	0,229	0,205	0,189	0,183	0,281	0,249	0,222	0,205	0,199	0,310	0,274	0,245	0,226	0,219
	H:	9,4	7,8	6,1	4,6	4,0	8,9	7,4	5,7	4,4	3,6	8,5	7,1	5,5	4,1	3,4	8,2	6,8	5,2	3,9	3,2
25	P:	0,216	0,193	0,176	0,164	0,160	0,233	0,209	0,190	0,177	0,173	0,254	0,228	0,207	0,193	0,189	0,281	0,251	0,228	0,213	0,208
	K:	0,235	0,208	0,186	0,172	0,167	0,254	0,224	0,201	0,185	0,179	0,275	0,243	0,218	0,201	0,195	0,303	0,268	0,240	0,221	0,215
	H:	9,2	7,6	5,9	4,6	3,8	8,7	7,3	5,6	4,3	3,5	8,3	6,9	5,3	4,0	3,3	8,0	6,7	5,1	3,8	3,1

és

$$dH_k(x) = k\binom{n-1}{k}\frac{1}{n(1-\lambda)}\left[\frac{x-k\lambda}{n(1-\lambda)}\right]^{k-1}\left[1-\frac{x-k\lambda}{n(1-\lambda)}\right]^{n-k-1}dx.$$

A $z=\frac{x-k\lambda}{n(1-\lambda)}$ helyettesítéssel a bizonyítandó (3.7) összefüggést nyerhetjük.

A (3.7) kifejezés az M változó monoton növekvő függvénye, így a minimális induló készletszint értéke, mely a [0, 1] intervallumban legalább $1-\varepsilon$ valószínűséggel biztosítja a folyamatos anyagellátást a

$$P\left(\sup_{0\leq t\leq 1}\{t-F_n(t,\lambda)\}\leqq M\right)=1-\varepsilon \tag{3.8}$$

egyenlet megoldása. Induló megoldásként a PRÉKOPA A. által megadott (1.4) közelítő megoldást választva a szokásos iterációs egyenletmegoldó módszerek bármelyikével néhány lépésben a gyakorlat számára elegendően pontos megoldást kaphatunk a (3.7) kifejezést felhasználva.

A (3.8) egyenlet három tizedes jegyre pontos megoldását táblázatba foglaljuk különböző ε, λ és n értékekre. Ez a gyakorlatban közvetlenül felhasználható. Az (1.4) közelítő megoldást és ennek relatív százalékos hibáját is tartalmazza az 1. táblázat, melyben a (3.8) egyenlet pontos (P), közelítő (K) megoldása, és a közelítés relatív százalékos hibája (H) szerepel.

4. Az induló készletszint meghatározása, ha az igény is véletlen

A folyamatos anyagellátást biztosító megfelelő készletszint kialakítása mindig az eggyel korábbi periódusra történő rendelés alapján lehetséges, ezért az induló készletszintet egy periódussal előre meg kell tervezni és ez alapján kell rendelni. A rendelési periódus hossza gyakran olyan nagy, hogy egy periódussal előre az anyagfelhasználás várható intenzitására csak durva becslés adható. Ebben az esetben mind az igény, mind a beérkezés folyamatának véletlen jellegét figyelembe kell venni az induló készletszint tervezésénél, mert az egyik folyamatra sem rendelkezünk elegendő pontos információval a döntéshozatal (rendelés) időpontjában.

Az igény α intenzitását valószínűségi változónak tekintjük, eloszlásfüggvényét $G(x)$ jelöli. Az $\eta(t)$ beérkezési folyamat esetén a szükséges induló készletszintet az

$$\int_0^\infty P\left(\sup_{0\leq t\leq 1}\{xt-\eta(t)\}\leqq M\,|\,\alpha=x\right)dG(x)=1-\varepsilon \tag{4.1}$$

egyenlet megoldása adja, melyet ismert numerikus módszerekkel meg tudunk oldani az $\eta(t)$ folyamat (1.1) vagy (1.2) szerinti modellje esetén, felhasználva az előző pont eredményeit.

A (4.1) egyenlet megoldása viszonylag sok számítást igényel, ezért csak a legfontosabb, illetve a legdrágább anyagok esetén érdemes ezt a pontos megoldást

meghatározni. A kevéssé fontos anyagok készletszintjének tervezésére egy egyszerű közelítő formulát kívánunk adni. A beérkezési folyamatot az $F_n(t)$ empirikus eloszlásfüggvénnyel közelítjük, így a CSÁKI E. [2] által adott következő aszimptotikus eloszlást alkalmazhatjuk:

$$\lim_{n\to\infty} P\left(\sup_{0\leq t\leq 1}\left\{\left(1+\frac{h}{\sqrt{n}}\right)t-F_n(t)\right\}\leqq\frac{v}{\sqrt{n}}\right)=1-e^{-2v(v-h)} \tag{4.2}$$

$v>0$ és $v>h$ esetén. Az $\alpha=1+h/\sqrt{n}$ és $M=v/\sqrt{n}$ helyettesítéssel n véges értékeire a következő közelítés adható

$$P\left(\sup_{0\leq t\leq 1}\{\alpha t-F_n(t)\}\leqq M\right)\approx 1-e^{-2nM(M+1-\alpha)} \tag{4.3}$$

a $\max\{0,\alpha-1\}<M<\alpha$ értékekre. Itt α jelöli a beérkezési periódusban igényelt és a beérkező mennyiség hányadosát, mint korábban. A (4.1) egyenlet megoldása α ismert értéke esetén a következő egyszerű formulával közelíthető

$$M\approx\frac{\alpha-1}{2}+\sqrt{\left(\frac{\alpha-1}{2}\right)^2+\frac{1}{2n}\ln\frac{1}{\varepsilon}}. \tag{4.4}$$

Ha α valószínűségi változó a (4.1) egyenlet baloldala az

$$1-e^{-2nM(M+1)}E[e^{2n\alpha M}] \tag{4.5}$$

alakban közelíthető, ahol E jelöli a várható értéket. Az

$$1-e^{-2nM(M+1)}E[e^{2n\alpha M}]=1-\varepsilon \tag{4.6}$$

egyenlet megoldása adja azt az M induló készletet, mely az előírt $1-\varepsilon$ valószínűségi szintet jól közelíti. Az α valószínűségi változó adott eloszlása esetén a (4.6) egyenlet numerikus módszerrel megoldható.

A gyakorlat számára legfontosabb esetben, amikoris a becslési hibát normális eloszlásúnak tételezzük fel, a (4.6) egyenlet megoldását explicit formában ki tudjuk fejezni. Jelölje az α normális eloszlású valószínűségi változó várható értékét m, szórását pedig s. Pozitív konstans c esetén a következő átalakítást végezhetjük:

$$E[e^{c\alpha}]=\frac{1}{\sqrt{2\pi}s}\int_{-\infty}^{\infty}e^{cx-\frac{(x-m)^2}{2s^2}}dx=e^{mc+\frac{c^2s^2}{2}}\frac{1}{\sqrt{2\pi}s}\int_{-\infty}^{\infty}e^{\frac{[x-(m+cs^2)]^2}{2s^2}}dx,$$

ahol a második tényező, mint a normális eloszlásfüggvény végtelenben vett értéke egyenlő eggyel. Így a $c=2nM$ helyettesítéssel a (4.3) közelítést felhasználva a

$$P\left(\sup_{0\leq t\leq 1}\{\alpha t-F_n(t)\}\leqq M\right)\approx 1-e^{-2nM[M+1-(m+nMs^2)]}$$

közelítést nyerhetjük a folyamatos ellátás valószínűségére. Ezt $1-\varepsilon$-nal egyenlővé téve, a kívánt ellátási valószínűséget közelítően biztosító induló készletszint a követ-

kező egyszerű formulával megadható:

$$M \approx \frac{m-1}{2(1-ns^2)} + \sqrt{\left[\frac{m-1}{2(1-ns^2)}\right]^2 + \frac{1}{2n(1-ns^2)} \ln \frac{1}{\varepsilon}} \tag{4.7}$$

az $s < 1/\sqrt{n}$ feltétel mellett. Itt m jelöli az α várható értékét, s pedig a szórását, mely paraméterek meghatározására a szokásos előrejelzési módszerekben mindig sor kerül.

5. A modellek alkalmazása

A Dunai Vasmű megbízásából egy készletgazdálkodási döntéselőkészítő programrendszer készült az MTA Számítástechnikai és Automatizálási Kutató Intézetének Operációkutatási Osztályán. A programrendszer egészének ismertetése GÖMBÖCZ L.—KELLE P.—SEBŐ A. [4] cikkben szerepel. Statisztikai előrejelzési módszereknek és készlet tervezési eljárásoknak egy moduláris felépítésű programrendszere ez, mely alkalmas a várható igények előrejelzésére, a biztonsági készletszint (készletnorma) tervezésére és a rendelési javaslat elkészítésére. A készlettervezés moduljai az irodalomban közölt, illetve ezen cikkben ismertetett megbízhatósági készletmodelleken alapulnak és sokféle típusú beérkezési, valamint felhasználási folyamat esetén alkalmazhatók.

A cikkben közölt modellek konstrukciója a Pamutnyomóipari Vállalat és a Dunai Vasmű számítógépes készletnyilvántartási rendszerének vizsgálata alapján indult. Mindkét rendszer rögzíti és több évig mágnesszalagokon megőrzi a raktári mozgásokat, a beérkezések és kivételezések időpontját és tételnagyságát. Ez az információ lehet az alapja a korábbiaknál pontosabb sztochasztikus modellek illesztésének minden egyes vizsgált termék esetén. A rendelésfeladás időpontjában mind a tervezendő időszak ellátási feltételei, mind várható anyagfelhasználása sok esetben csak durván becsülhető. Nagy tömegű anyag és termékféleség esetén sokszor csak a korábbi periódusok archivált adatai használhatók fel a számítógépes döntéselőkészítésre. Erre épülnek a statisztikai becslési eljárások, melyekkel a modell illesztések elvégezhetők.

Két egymást követő rendelési periódus anyagfelhasználási terve vagy ennek hiányában az igény előrejelzése alapján és az induló készletszint egy periódussal történő előre tervezése alapján készül a rendelési javaslat, mely a megfelelő záró készlet kialakítását és így az azt követő periódus zavartalan anyagellátást biztosítja. A legfontosabb anyagféleségekre az induló készletszint tervezése a (4.1) eljárás alapján kipróbálásra került. A nagy tömegű anyag- és termékféleség (a Dunai Vasmű esetében közel 60 000) szükségessé teszi azt, hogy a gyors működés érdekében egyszerű közelítő megoldásként az (1.4), (4.4) és (4.7) formulákat alkalmazzuk, megfelelő numerikus korrekcióval.

A programrendszer több anyagcsoporton (közel 10 000 anyag- és termékféleségre) való tesztelése jó eredményeket hozott. Egy elegendően magas ellátási biztonsági szint mellett a teljes készletmennyiség csökkentésének lehetőségét igazolta, mely jelentős költségmegtakarítást eredményezhet. A készletnormák korrigálása a programrendszer alapján lényegében elkészült minden anyag- és termékféleségre. A rendelési javaslat készítési rendszer üzemszerű beindítása folyamatban van.

IRODALOM

[1] BIRNBAUM, Z. W. and TINGEY, H. F., "One-sided confidence contours for probability distribution function" *Annals of Math. Statistics* **22** (1951) 592—596.

[2] CSÁKI, E., „Empirikus eloszlásfüggvényekkel kapcsolatos vizsgálatok", kandidátusi értekezés, Budapest, 1974.

[3] GERENCSÉR, L., "Reduction of the on-hand inventory on given level of reliability", Prékopa A. szerk. *Colloquia Math. Soc. J. Bolyai 7. Inventory Control and Water Storage Győr,* 1971 (Bolyai J. Math. Soc. and North Holland Publ. Comp. Budapest 1973.) 83—92.

[4] GÖMBÖCZ, L., KELLE P. és SEBŐ A., „Készletgazdálkodási döntéselőkészítő programrendszer a Dunai Vasműben", *Struktúra* 1981/14 61—82.

[5] KELLE, P., „Stochastische Mehr-Produkt Modelle für die Sicherheitsbestände bei einer Serienfabrikation", *Rostocker Betriebswirtschaftliche Manuscripte* **20** (1977) 151—166.

[6] KELLE, P., „Stochastische Optimierungsmodelle für die Produktionslager eines Walzwerkes", *Mitteilungen der Math. Gesellschaft d. DDR* 1978/1. 31—36.

[7] KELLE, P., „Megbízhatósági készletmodellek és alkalmazásuk" *MTA SZTAKI Tanulmányok* 107 (1980).

[8] LÁSZLÓ, Z., „Egy teljesen véletlen megbízhatósági jellegű készletmodell", kandidátusi értekezés, Veszprém, 1970.

[9] LÁSZLÓ, Z., „Megbízhatósági készletmodellekkel kapcsolatos eredmények" előadás, IX. Magyar Operációkutatási Konferencia, Szeged, 1978.

[10] MÓRITZ, A., „Ein mathematisches Modell für die Lagerhaltung in Ungarischen Binnenhandel". *Zeitschrift für Operations Research* **22** (1978) 345—56.

[11] NÉMETH, GY., „Sztochasztikus készletmodellekkel kapcsolatos vizsgálatok", *MTA III. Oszt. Közlemények* **16** (1971) 133—135.

[12] PINTÉR, J., „Egy sztochasztikus irányítási feladat megoldása és alkalmazása egy készletmodellre", *Egyetemi Számítóközpont Közleményei,* 1974.

[13] PINTÉR, J. „Empirikus eloszlásfüggvény-sorozatok maximális eltérésének vizsgálata: alkalmazás egy több periódusú megbízhatósági készletmodellre", *Alk. Mat. Lapok* **1** (1975) 189—195.

[14] PRÉKOPA, A., "Reliability equation for an inventory problem and its asymptotic solution", in: *Coll. on Appl. of Math. to Economics* (Akadémia Kiadó, Budapest, 1965) 317—327.

[15] PRÉKOPA, A., "Stochastic Programming Models for Inventory Control and Water Storage Problems", in: *Colloquia Math. Soc. J. Bolyai 7. Inventory Control and Water Storage Győr,* 1971. (Bolyai J. Math. Soc. and North Holland Publ. Comp., Budapest, 1973). 229—246.

[16] PRÉKOPA, A., "Generalizations of the Theorems of Smirnov with Application to a Reliability Type Inventory Problem", *Mathematische Operationsforschung und Statistic* **4** (1973) 283—297.

[17] PRÉKOPA, A. és KELLE, P., „Sztochasztikus programozáson alapuló megbízhatósági jellegű készletmodellek", *Alk. Mat. Lapok* **2** (1976) 1—16.

[18] RÉNYI, A., *Valószínűségszámítás* (Tankönyvkiadó, Budapest, 1966.).

[19] SMIRNOV, N., "On the estimation of the discrepancy between empirical curves of distribution for two independent samples", *Bull. Math. Univ. Moscow* **2** (1939) 3—16.

[20 TAKÁCS, L., *Combinatorial Methods in the Theory of Stochastic Processes* (J. Wiley, New York, 1967.)

[21] VASS, I., "Application of an inventory model based on a stochastic input-output process in the distribution industry", *Kerinforg,* 1973.

[22] WILKS, P. S., *Mathematical Statistics* (J. Wiley, New York, 1967).

[23] ZIERMANN, M., „A Szmirnov-tétel alkalmazása egy raktározási problémára", *MTA Matematikai Kutató Intézetének Közleményei* **8** (1963) 509—516.

(Beérkezett: 1981. szeptember 15.)

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

TWO EXTENSIONS OF A RELIABILITY TYPE INVENTORY MODEL OF PRÉKOPA

P. KELLE

The first inventory model in which the delivery of an order accure in random time points and in random amounts has been formulated by A. PRÉKOPA [14]. He has derived a simple approximate formula for the planning of the initial stock which serves as safety stock and ensures the material supply of continuous production in the order period on a prescribed probability level. This model which has been widely used in practice will be extended in this paper. First an arbitrary distribution is allowed for the random delivery amounts, thus it can be approximated by the statistical data of earlier observations available in the management system for stock control. We derive the probability of the continuous material supply which can be applied for the exact solution of the original model, too. The model will be extended also for the case when the demand is a random variable. For normal distributed forecasting error a simple approximate formula is given for the necessary initial stock level. The above models are built in an inventory control program package developed by the operations research department of the Computer and Automation Institute of the Hungarian Academy of Sciences which has been applied in the Danubian Iron Works.

Alkalmazott Matematikai Lapok 7 (1981) 205—216

AZ ÁLTALÁNOS ELOSZTÁSI (SZÁLLÍTÁSI) FELADAT MEGOLDÁSÁNAK ALGORITMUSA A SOR- ÉS OSZLOP-SAJÁTELEM FOGALOM FELHASZNÁLÁSÁVAL

GYURKÓ GYÖRGY

Salgótarján

Ez a cikk egy egyszerű algoritmust mutat be az általános szállítási (elosztási) feladat megoldására. Az algoritmus a báziselemek felett definiált sor- és oszlop-sajátelem fogalomra épül. A klasszikus szállítási feladat *Glover-féle három-címke módszerével* összehasonlítva ([1], [2]), ez az eljárás az általános feladat egy egy-címke módszerének tekinthető.

1. Alapfogalmak. A feladat leírása

Az elosztási (szállítási) probléma egy speciális lineáris programozási feladat. Sajátosságai lehetővé teszik, hogy a szimplex transzformációkat — amelyeknek tárgya egy $(m+n)\times(mn)$ méretű mátrix —, olyan gazdaságosabb operációkkal helyettesítsük, amelyeket egy jelentősen kisebb — $m\times n$ méretű — ún. disztribúciós táblán (továbbiakban: DT) folytatunk. Általában olyan műveletekről van szó, amelyeket a DT báziselemeiből — mint csomópontokból — álló gráfok felett definiálunk.

A megoldás elve — a disztribúciós eljárás — régóta egyszerű tananyag ([3], [5]), de gyakorlati megvalósítását megnehezíti, hogy a szóban forgó gráfok nem közvetlenül adottak, azokat valahogy elő kell állítani. Még kisméretű feladatok „kézi" megoldása esetén e gráfok azonnal szembetűnőek, de nagyobb feladatok számítógépi megoldásánál kijelölésük már igen komoly algoritmizálási probléma.

Szerencsére a feladat sajátosságaiból az is adódik, hogy a DT „útkereszteződéseiben" alkalmas „útjelző táblák" helyezhetők el. Az ilyen útjelzők szerepét *Glover módszerénél* (csak a klasszikus feladatra) a hármas indexelés *(predecessor index method, triple-label method* [1], [2]*)*, POGÁNY ZSUZSÁNÁL (az általános feladatra) a négyes indexelés ([4]) tölti be; ez azt jelenti, hogy a kérdéses gráfokat a DT soraihoz és oszlopaihoz rendelt index-hármasokkal, illetve index-négyesekkel írják le.

Az itt tárgyalt algoritmusban az útjelző funkciót a sor- és oszlop-sajátelem fogalom látja el. Itt sor- és oszlopindexelésről nem fogunk beszélni, de a korábbi megoldásokkal való összevetés céljából megjegyezzük: a sajátelem fogalommal végzett műveletek azt bizonyítják, mindig létezik a DT sorainak és oszlopainak olyan indexelése, hogy

— egyetlen indexet rendelünk minden sorhoz és oszlophoz,
— a disztribúciós eljárás megfogalmazható ezen egyes indexelés felett definiált könnyen programozható eljárások sorozataként.

Ezt az állítást az algoritmus ismeretében bárki beláthatja, ha a DT soraihoz a sorsajátelem oszlopindexét, oszlopaihoz az oszlop-sajátelem sorindexét rendeli.

A továbbiakban az algoritmust olyan általános elosztási feladaton mutatjuk be, amely egyébként normál lineáris programozási feladat; ez azonban nem jelenti azt,

hogy nem-normál feladatra nem volna kiterjeszthető. Mint ahogy a szimplex algoritmust nem-normál feladat esetében két fázisban — előbb egy pótcélfüggvényre, majd az eredeti célfüggvényre — alkalmazzuk, pontosan úgy ennek a módszernek is megkonstruálható a kétfázisú változata. Összefoglalva: ez az eljárás az általános elosztási feladatok vonatkozásában az alkalmazhatóság tekintetében a szimplex algoritmussal ekvivalens, de annál rövidebb úton vezet a célhoz.

A feladat:

$$
\begin{aligned}
& x_{ij} \geqq 0, && i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n, \\
& x_{m+1,j} \geqq 0, && j = 1, 2, \ldots, n, \\
& x_{i,n+1} \geqq 0, && i = 1, 2, \ldots, m, \\
& \sum_{j=1}^{n} f_{ij} \cdot x_{ij} + x_{i,n+1} = t_i, && i = 1, 2, \ldots, m; \quad t_i > 0, \\
& f_{ij} \neq 0, && i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n, \\
& \sum_{i=1}^{m+1} x_{ij} = r_j, && j = 1, 2, \ldots, n; \quad r_j > 0.
\end{aligned} \tag{1.1}
$$

$$
\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \cdot x_{ij} \to \text{maximum}.
$$

(f_{ij}, c_{ij}, t_i, r_j előre adott konstansok.)

Megjegyezzük, hogy az (1.1) feladatnak — lévén normál lineáris programozási probléma — mindig van lehetséges bázismegoldása.

Disztribúciós táblán (DT) az (1.1) feladat együtthatóinak az (1.2) táblázat szerinti elrendezését értjük.

Megkülönböztetett szerepe miatt a DT $(m+1)$-edik sorát fiktív sornak (FS), a DT $(n+1)$-edik oszlopát fiktív oszlopnak (FO) nevezzük. Ennek megfelelően beszélni fogunk FS-, illetve FO-elemekről.

(1.2)

f_{11} c_{11}	...	f_{1j} c_{1j}	...	f_{1n} c_{1n}	$(f_{1,n+1}=)1$ $0(=c_{1,n+1})$	t_1
...	...	...	...	...	...	...
f_{i1} c_{i1}	...	f_{ij} c_{ij}	...	f_{in} c_{in}	$(f_{i,n+1}=)1$ $0(=c_{i,n+1})$	t_i
...	...	...	...	...	...	...
f_{m1} c_{m1}	...	f_{mj} c_{mj}	...	f_{mn} c_{mn}	$(f_{m,n+1}=)1$ $0(=c_{m,n+1})$	t_m
0 $0(=c_{m+1,1})$	...	0 $0(=c_{m+1,j})$	...	0 $0(=c_{m+1,n})$		
r_1	...	r_j	...	r_n		

A további meggondolások szempontjából érdemes a feladat következő mátrixleírására is figyelemmel lenni:

$$\mathbf{L} = \begin{bmatrix} \mathbf{A} & \mathbf{E} & \mathbf{b} \\ \mathbf{c}^* & \mathbf{0}^* & 0 \end{bmatrix}, \tag{1.3}$$

ahol

$$\mathbf{a}_{ij} = f_{ij}\mathbf{e}_i + \mathbf{e}_{m+j}, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n;$$

$$\mathbf{A} = [\mathbf{a}_{11}, \mathbf{a}_{12}, \ldots, \mathbf{a}_{1n}, \mathbf{a}_{21}, \ldots, \mathbf{a}_{2n}, \ldots, \mathbf{a}_{m1}, \ldots, \mathbf{a}_{mn}, \tag{1.4}$$

$$\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{m+n}] = \tag{1.5}$$

$$= [\mathbf{a}_{1,n+1}, \mathbf{a}_{2,n+1}, \ldots, \mathbf{a}_{m,n+1}, \mathbf{a}_{m+1,1}, \mathbf{a}_{m+1,2}, \ldots, \mathbf{a}_{m+1,n}],$$

$$\mathbf{c}^* = [c_{11}, c_{12}, \ldots, c_{1n}, c_{21}, \ldots, c_{2n}, \ldots, c_{m1}, \ldots, c_{mn}]. \tag{1.6}$$

$$\mathbf{b} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_m \\ r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}. \tag{1.7}$$

Az (1.5) szerint fennáll:

$$\mathbf{a}_{ij} = f_{ij}\mathbf{e}_i + \mathbf{e}_{m+j} = f_{ij}\mathbf{a}_{i,n+1} + \mathbf{a}_{m+1,j}; \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n. \tag{1.8}$$

Beszélni fogunk a DT vektorrendszeréről, illetve bázisáról, ezen az $[\mathbf{A}, \mathbf{E}]$ mátrix oszlopvektorainak rendszerét, illetve annak valamely bázisát kell érteni. Ennek megfelelően a továbbiakban nem is fogunk mindig különbséget tenni a DT (i, j) eleme és a hozzá tartozó $\mathbf{a}_{ij}$ vektor között. Ezekután (1.5) alapján elmondható, hogy a DT vektorrendszerének rangja $m+n$.

A disztribúciós eljárás itt leírt módozatának legfontosabb mozzanatai az egyes transzformációs lépések alkalmával:

DE1. A sajátelemek kijelölése.
DE2. A potenciálok meghatározása.
DE3. A célfüggvényegyütthatók transzformált értékeinek meghatározása.
DE4. A bázismegoldás optimális vagy nem optimális voltának eldöntése, javításra szoruló megoldás esetén a bázisba bevonandó elem kiválasztása.
DE5. A bázisba bevonandó vektor (aktuális bázisra vonatkozó) komponenseinek meghatározása és a szűk keresztmetszet kiválasztása. (Kompenzációs eljárás.)
DE6. A transzformáció elvégzése.

2. A sajátelemek kijelölése

A korábbi megoldásokhoz képest itt a sajátelem fogalom jelenti a leglényegesebb különbséget. E fogalomra építve a disztribúciós eljárás szinte minden mozzanata egyszerűsödik. A sajátelem fogalom matematikailag lényegében egy a DT bázis-elemein értelmezett függvényt jelent, és mint ilyet, éppen a hozzárendelési algoritmussal fogjuk definiálni.

Magának a hozzárendelésnek kétféle felfogása lehetséges: egyszer úgy tekinthető, mint a báziselemek halmaza, valamint a sor-sajátelem és oszlop-sajátelem tulajdonságok kételemű halmaza közti hozzárendelés; máskor pedig tekinthető, mint a bázis-elemek halmaza és a DT sorainak és oszlopainak halmaza közti hozzárendelés. A további tárgyalás során a két felfogás közül célszerűen hol egyik, hol másik kerül majd előtérbe.

Megjegyezzük még, hogy a sajátelemek kijelölésének itt következő S1—S7 algoritmusa nem egyértelmű, amennyiben az S7 pontban önkényes választásra ad lehetőséget. Ez azonban nem lehet hiba forrása, mert bebizonyítható, hogy a disztribúciós eljárás további mozzanatai közömbösek aziránt, hogy korábban az S7 pontban milyen választással éltünk.

Legyen adott a DT egy bázisa (ez mindig $m+n$ elemből áll). A DT báziselemei felcímkézhetők a *sor-sajátelem* (SS), illetve az *oszlop-sajátelem* (OS) elnevezésekkel úgy, hogy

— minden báziselem felcímkéződik;
— egy báziselem vagy csak SS-elem lehet, vagy csak OS-elem;
— a DT minden nem-fiktív sorában pontosan egy SS-elem van;
— a DT minden nem-fiktív oszlopában pontosan egy OS-elem van;

A felcímkézés (=kijelölés) a következő algoritmus szerint történhet:

S1. A DT FO-beli báziselemeit jelöljük ki SS-elemnek.
S2. A DT FS-beli báziselemeit jelöljük ki OS-elemnek.
S3. Minden nem-fiktív sorban, amelyben már van SS-elem, az összes még jelöletlen báziselemet jelöljük ki OS-elemnek.
S4. Minden nem-fiktív oszlopban, amelyben már van OS-elem, az összes még jelöletlen báziselemet jelöljük SS-elemnek. Ezután, ha van még jelöletlen báziselem valamely SS-elem sorában, akkor menjünk az S3 pontra.
S5. Minden nem-fiktív sorban, amelyben még nincs SS-elem, és csak egy kijelöletlen báziselemet tartalmaz, ezt a jelöletlen báziselemet jelöljük SS-elemnek.
S6. Minden nem-fiktív oszlopban, amelyben még nincs OS-elem, és csak egy jelöletlen báziselemet tartalmaz, ezt a jelöletlen elemet jelöljük OS-elemnek. Végül, ha van még olyan nem-fiktív sor, amelyben nincs SS-elem, és pontosan egy jelöletlen báziselemet tartalmaz, akkor menjünk az S5 pontra.
S7. Ha az eddigiekben nem jelöltünk ki minden báziselemet, akkor a jelöletlenek közül egyet vegyünk SS-elemnek, és menjünk az S3 pontra.

Megjegyzés: Ha az algoritmus egy pontja másképpen nem rendelkezik, akkor azt mindig a következő pont követi.

Az S1—S7 pontok végrehajtására igen könnyen igen hatékony számítógépes program írható.

Az, hogy a báziselemek nem helyezkedhetnek el akárhogyan a DT-ben, de mindig úgy helyezkednek el, hogy az előbbi algoritmus elvégezhető, és a sajátelemek felsorolt tulajdonságai mindig teljesülnek; a bázisvektorrendszer lineáris függetlenségére, az (1.8) egyenlőségekre, valamint a DT sajátos szerkezetére támaszkodva bizonyítható be.

3. A potenciálok meghatározása, a célfüggvényegyütthatók transzformált értékeinek számítása

A potenciálok meghatározása azt jelenti, hogy olyan u_i, illetve v_j skalárokat rendelünk a DT nem-fiktív soraihoz, illetve oszlopaihoz, amelyekkel a c_{ij} célfüggvény együtthatók transzformált értékei ($\hat{c}_{ij}$) így számíthatók:

C1. Minden nem-fiktív (i, j) esetén

$$\hat{c}_{ij} = c_{ij} - (f_{ij} u_i + v_j).$$

C2. Minden $(i, n+1)$ esetén

$$\hat{c}_{i,n+1} = -u_i.$$

C3. Minden $(m+1)$ esetén

$$\hat{c}_{m+1,j} = -v_j.$$

Ilyen potenciálok egyértelműen megadhatók, figyelembe véve:

C4. Minden (i, j) báziselem esetén: $\hat{c}_{ij} = 0$.

Az általános elosztási feladat esetében a báziselemek hurkot is alkothatnak, ami a potenciálok meghatározása szempontjából különleges körülmény. *Huroknak* nevezzük a DT bázisvektorainak olyan nem-üres halmazát, amely a DT minden sorából és oszlopából — ha onnan tartalmaz elemet — pontosan két báziselemet tartalmaz; viszont nincs olyan valódi részhalmaza, amelyre szintén igaz lenne ez a tulajdonság.

Az elosztási feladatra adott eddigi megoldásoknál is felvetődött néhány kérdés:

K1. Milyen kritérium alapján tudjuk eldönteni, hogy a DT-ben léteznek-e hurkok vagy sem?

K2. Ha létezik hurok, akkor annak az elemei hogyan kereshetők fel a DT-ben?

K3. Végül, ha adott a hurok az elemeivel, akkor hogyan határozhatók meg az annak soraihoz és oszlopaihoz tartozó potenciálok?

A K1-re adott válasz: Ha a sajátelemek kijelölése során alkalmazni kellett az S7 pontot, akkor a DT-ben van hurok, és pontosan annyi hurok van benne, ahányszor az S7-et alkalmaztuk.

A K2 és K3 kérdésekre a választ a *hurok-algoritmus* adja meg. Ennek lényege:

HU1. Az S7 pontban meghatározott SS-elem hozzá tartozik valamely hurokhoz, tehát a hurok elemeinek megkeresése vele kezdhető. Legyenek az indexei: i_0, j_1. Innen megyünk az SS-elem oszlopában levő OS-elemhez (legyenek indexei: i_2, j_1), innen megyünk az OS-elem sorában levő SS-elemhez (legyenek indexei: i_2, j_3), és így tovább, míg végül a kiinduló SS-elem sorában levő OS-elemhez nem jutunk (legyenek indexei: i_s, j_{s-1}, ahol s páros).

HU2. A HU1 pontban kijelölt hurok elemeit a kijelölés sorrendjében végigjárva generálunk két véges számsorozatot:

$$p_0, p_1, \ldots, p_s,$$

$$q_0, q_1, \ldots, q_s.$$

A sorozatok definíciója:

$$p_0 = 0, \qquad q_0 = 1,$$

$$p_{2k} = \frac{c_{i_{2k}, j_{2k-1}} - p_{2k-1}}{f_{i_{2k}, j_{2k-1}}}, \qquad q_{2k} = \frac{-q_{2k-1}}{f_{i_{2k}, j_{2k-1}}},$$

$$p_{2k+1} = c_{i_{2k}, j_{2k+1}} - f_{i_{2k}, j_{2k+1}} p_{2k}, \quad q_{2k+1} = -f_{i_{2k}, j_{2k+1}} q_{2k}.$$

(Számítástechnikai szempontból lényeges, hogy végül a $\{p_i\}$, $\{q_i\}$ sorozatokból csak az utolsó (s-edik) elemekre van szükség. — Tehát, például p_{i-1}-et csak addig szükséges tárolni, amíg p_i-t nem határoztuk meg.)

HU3. Az (i_0, j_1)-hez tartozó u_{i_0} potenciál így állítható elő:

$$u_{i_0} = \frac{p_s}{1 - q_s}.$$

Ezekután a potenciálok meghatározását így foglalhatjuk össze:

P1. Minden fiktív SS-elem sorához rendeljük: $u_i = 0$.

P2. Minden fiktív OS-elem oszlopához rendeljük: $v_j = 0$.

P3. Minden olyan SS-elem esetén, amelyet az S7 pontban jelöltünk, alkalmazzuk a *hurok-algoritmust*.

P4. Minden olyan SS-elem esetén, amelynek oszlopához már tartozik v_j (de a sorához még nem tartozik u_i), a sorához rendeljük:

$$u_i = \frac{c_{ij} - v_j}{f_{ij}}. \tag{3.1}$$

P5. Minden olyan OS-elem esetén, amelynek sorához már tartozik u_i (de oszlopához még nem tartozik v_j) rendeljük:

$$v_j = c_{ij} - f_{ij} u_i. \tag{3.2}$$

Ha végül maradt még határozatlan potenciál, akkor térjünk vissza a P4 ponthoz.

Megjegyzések. A P4, illetve P5 pontban i, j az SS-, illetve OS-elem indexei. Ha az algoritmus egy pontja másképpen nem rendelkezik, akkor azt mindig a következő pont követi.

Ezzel túl vagyunk az algoritmus DE1—DE3 pontjain. A DE4 pont a szimplex algoritmusnál leírtak szerint történhet.

A fejezetben leírtak bizonyíthatósága. A C1—C4 összefüggések a lineáris programozás elméletéből ismert tételek feladatunkra adaptálásából adódnak. A hurok létezésének kritériuma (válasz K1-re), valamint megkeresésének módja (HU1 pont)

a DT sajátosságaira, a hurok tulajdonságaira és a bázisvektorok lineáris függetlenségére támaszkodva bizonyítható be. A HU2, HU3 és a P1—P5 pontok a C1—C4 pontokból következnek. Közelebbről: a HU2, HU3 pontokat megkapjuk, ha a hurok báziselemeire a C1 és C4 alapján felírható lineáris egyenletrendszert a *Gauss-módszerrel* úgy oldjuk meg, hogy közben az egyenleteket a nekik megfelelő báziselemek HU1 pont szerinti sorrendjében vesszük figyelembe.

4. A bázisba bevonandó vektor aktuális bázisra vonatkozó komponenseinek meghatározása

A bázisba bevonandó elem aktuális bázisra vonatkozó komponenseinek meghatározásához először kijelölünk bizonyos gráfo(ka)t a DT-ben, amely(ek) a DT báziselemeiből áll(nak).

Részletesen:

K1. Ha a bevonandó elem nem fiktív, akkor hozzárendelünk egy ún. *oszloputat* és egy *sorutat* DT-ben. — E fogalmakat éppen a hozzárendelési algoritmus definiálja (lásd a következőkben).

K2. Ha a bevonandó elem fiktív oszlophoz tartozik, akkor egy sorutat rendelünk hozzá.

K3. Ha a bevonandó elem fiktív sorhoz tartozik, akkor egy oszloputat rendelünk hozzá.

A sorút (oszlopút) meghatározásához előbb definiáljuk a hurokpont és a csomó fogalmát. Legyen $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \ldots$, a DT vektorainak (elemeinek) egy sorozata, és legyen j a legkisebb olyan index, amelyre $\mathbf{b}_j = \mathbf{b}_i$ és $0 \leqq i < j$. Ekkor $\mathbf{b}_j$-t a sorozat *hurokpontjának* nevezzük.

Ha valamely $\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_j, \ldots$ sorozatnak a $\mathbf{b}_j$ hurokpontja, akkor e *sorozat által generált csomón* a $\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_{j-1}, \mathbf{b}_{i-1}, \mathbf{b}_{i-2}, \ldots, \mathbf{b}_0$ sorozatot értjük.

Most legyen adott a DT egy bázisa és rajta a sajátelemek egy kijelölése. Legyen $\mathbf{b}_0$ a DT egy olyan vektora, amely nem tartozik az adott bázishoz és nem FS-elem. Ekkor a $\mathbf{b}_0$-hoz tartozó sorutat a következőképpen jelöljük ki.

UT1. Legyen $\mathbf{b}_1$ a $\mathbf{b}_0$-sorában levő SS-elem; ha ez fiktív, akkor ez a sorút vége; különben a folytatás az UT2 pontban.

UT2. Ha már adott a sorút a $\mathbf{b}_j$ SS-elemig, akkor legyen $\mathbf{b}_{j+1}$ a $\mathbf{b}_j$ oszlopában levő OS-elem. Ha ez fiktív, akkor ez az út vége Ha ez hurokpont, akkor következzék az UT4. Egyébként folytassuk az UT3 szerint.

UT3. Ha már adott az út a $\mathbf{b}_j$ OS-elemig, akkor legyen $\mathbf{b}_{j+1}$ a $\mathbf{b}_j$ sorában levő SS-elem. Ha ez fiktív, akkor ez az út vége. Ha ez hurokpont, akkor következzék az UT4. Különben folytassuk az UT2 szerint.

UT4. Ha már adott egy sorozat a $\mathbf{b}_j$ hurokpontig, akkor legyen az út a sorozat által generált csomó.

Az oszlopút csak abban különbözik a sorúttól, hogy a $\mathbf{b}_0$-ból OS-elemhez indul.

Miután meghatároztuk a sorutat és/vagy az oszloputat, ezek minden $\mathbf{b}_k$ báziseleméhez rendelünk egy c_k kompenzációs együtthatót. A rövidség kedvéért itt csak a sorút esetét tárgyaljuk.

Fiktív báziselemet tartalmazó sorút esetén a kompenzációs együtthatókat (c_k) a következő hozzárendeléssel definiáljuk:

$$\mathbf{b}_1 = \mathbf{a}_{i_0, j_1} \to c_1 = \frac{f_{i_0, j_{-1}}}{f_{i_0, j_1}};$$

(Itt i_0, j_{-1} a bevonandó elem indexei.)

$$\begin{aligned} \mathbf{b}_2 &= \mathbf{a}_{i_2, j_1} \to c_2 = -c_1; \\ \mathbf{b}_{2k} &= \mathbf{a}_{i_{2k}, j_{2k-1}} \to c_{2k} = -c_{2k-1}; \\ \mathbf{b}_{2k+1} &= \mathbf{a}_{i_{2k}, j_{2k}+1} \to c_{2k+1} = -\frac{f_{i_{2k}, j_{2k-1}}}{f_{i_{2k}, j_{2k+1}}} \cdot c_{2k}. \end{aligned} \tag{4.1}$$

Fiktív báziselemet nem tartalmazó sorút esetén szintén egy c_i kompenzációs együtthatót rendelünk a sorút minden $\mathbf{b}_i$ báziseleméhez, de ezt megelőzi egy $\mathbf{b}_i \to d_i$ hozzárendelés.

$$\begin{aligned} \mathbf{b}_1 &= \mathbf{a}_{i_0, j_1} \to d_1 = \frac{f_{i_0, j_{-1}}}{f_{i_0, j_1}}; \\ \mathbf{b}_2 &= \mathbf{a}_{i_2, j_1} \to d_2 = -d_1; \\ \mathbf{b}_{2k} &= \mathbf{a}_{i_{2k}, j_{2k-1}} \to d_{2k} = -d_{2k-1}; \\ \mathbf{b}_{2k+1} &= \mathbf{a}_{i_{2k}, j_{2k+1}} \to d_{2k+1} = -\frac{f_{i_{2k}, j_{2k-1}}}{f_{i_{2k}, j_{2k+1}}} \cdot d_{2k}. \end{aligned} \tag{4.2}$$

Ha most $\mathbf{b}_s$ a sorút utolsó báziseleme, akkor legyen

$$\alpha = \frac{f_{i_0, j_{-1}}}{f_{i_0, j_1} \cdot d_1 + f_{i_s, j_{s-1}} \cdot d_s}, \tag{4.3}$$

és legyen $c_i = d_i \cdot \alpha$ $(i = 1, 2, \ldots, s)$.

Mármost legyen adott a DT egy bázisa és rajta a sajátelemek egy kijelölése. Legyen $\mathbf{b}_0$ a DT olyan vektora, amely nem tartozik az adott bázishoz. Ennek az adott bázisra vonatkozó komponenseit úgy határozzuk meg, hogy felírjuk a báziselemek lineáris kombinációjaként a következők szerint:

Jelölje s a sorút, q az oszlopút $\mathbf{b}_0$-on kívüli tagjainak számát; $\mathbf{b}_k$ a sorút, $\hat{\mathbf{b}}_k$ az oszlopút k-adik tagját; c_k a $\mathbf{b}_k$-hoz, $\hat{c}_k$ a $\hat{\mathbf{b}}_k$-hoz tartozó kompenzációs együtthatót. Ekkor:

ha $\mathbf{b}_0$ FO-*elem, úgy*

$$\mathbf{b}_0 = \sum_{k=1}^{s} c_k \mathbf{b}_k; \tag{4.4}$$

ha $\mathbf{b}_0$ FS-*elem, úgy*

$$\mathbf{b}_0 = \sum_{k=1}^{q} \hat{c}_k \hat{\mathbf{b}}_k; \tag{4.5}$$

ha $\mathbf{b}_0$ *nem fiktív elem, úgy*

$$\mathbf{b}_0 = \sum_{k=1}^{s} c_k \mathbf{b}_k + \sum_{k=1}^{q} \hat{c}_k \hat{\mathbf{b}}_k. \tag{4.6}$$

Igy, mostmár a $\mathbf{b}_0$ aktuális bázisra vonatkozó komponenseinek ismeretében, a szűk keresztmetszet meghatározása és a DE6 pont úgy végezhető, mint a szimplex algoritmusnál.

5. További lényeges egyszerűsítések

A sajátelemek kijelölésének S1—S7 algoritmusát és a potenciálok kijelölésének P1—P5 algoritmusát tekintsük alapalgoritmusoknak.

Belátható, hogy ezen algoritmusokat teljes terjedelmükben elegendő csupán egyetlenegyszer alkalmazni; azaz a legelső transzformációs lépésnél, amikoris még úgy adott a DT valamely állapota (a feladat valamely lehetséges bázismegoldása), hogy nem ismerünk egyetlen potenciált sem, nincs kijelölve még egyetlen sajátelem sem. Ha azonban valamely transzformációs lépésben már ezeket megadtuk, akkor a következő lépéshez tartozó sajátelemek, illetve potenciálok az előző lépéshez tartozókból egyszerűbben származtathatók.

A sajátelemek származtatása:

SS1. Ha a bázisból kieső elem a sorúthoz tartozik, de az oszlopúthoz nem tartozik (vagy nincs oszlopút), akkor a bázisba éppen bevont elem legyen SS-elem.

SS2. Ha a bázisból kieső elem az oszlopúthoz tartozik, de a sorúthoz nem tartozik (vagy nincs sorút), akkor a bázisba éppen bevont elem legyen OS-elem.

SS3. Ha bázisból kieső elem a sorúthoz, oszlopúthoz is hozzátartozik, akkor csak az egyik utat vegyük figyelembe, és aszerint az SS1 vagy SS2 pontot alkalmazzuk. (Célszerű a rövidebb utat figyelembe venni.)

SS4. Ha a bevonandó elem SS-elem lett, akkor onnan elindulva a sorúton a régi SS-elemeket változtassuk OS-elemmé, a régi OS-elemeket változtassuk SS-elemmé; egészen addig, amíg nem értünk a bázisból kiesett elemhez. A sorúthoz nem tartozó, illetve a sorúton a kiesett elemen túl fekvő báziselemek ugyanolyan sajátelemek maradnak, mint azelőtt.

SS5. Ha a bevonandó elem OS-elem lett, akkor onnan az oszlopúton elindulva végezzük az SS4 ponthoz hasonló változtatásokat.

Az S1—S7 pontok mellőzése, azonnal felveti a kérdést: ezután milyen kritérium szerint dönthető el, hogy hány hurok létezik DT-ben? Tény, hogy az S7 pontra épülő kritériumtól elesünk. Csakhogy a fenti kérdés eldöntése eljárásunk szempontjából teljesen közömbös. A lényeges kérdés: Az új báziselem bevonásával keletkezik-e új hurok? Érvényesek a következő állítások:

A1. Az új hurok keletkezésének szükséges és elegendő feltétele a következők együttes teljesülése: a bázisba bevonandó elem nem fiktív; a sorútnak és az oszlopútnak van közös tagja; ilyen közös tag esik ki a bázisból.

A2. Az új hurok mindig tartalmazza az új báziselemet.

Az egyes transzformációs lépések alkalmával általában a potenciáloknak csak egy része változik. Ha az új báziselem új hurkot határoz meg, akkor a hurokalgoritmus szerint a hurok soraihoz és oszlopaihoz tartozó új potenciálokat határozzuk meg. A HU1 pont annyiban módosul, hogy a kezdőelem megválasztás így történik:

— ha az új báziselem SS-elem, akkor legyen ez a hurok kezdő eleme,
— egyébként legyen az új báziselem sorában levő SS-elem.

Ezután a potenciálok számításának P4 és P5 pontja szerint járunk el (alapalgoritmus), azzal a változtatással, hogy a P5 pontból csak akkor térünk vissza a P4-hez, ha olyan SS-elem soránál határozatlan a potenciál, amelynek oszlopához már rendeltünk új potenciált. Ha végül maradnak olyan báziselemek, amelyekhez ez az eljárás nem rendelt potenciálokat, akkor azok sorában és oszlopában a régi potenciálokat hagyjuk meg.

Ha az új báziselemhez nem tartozik hurok, és az SS-elem, akkor a sorához tartozó új potenciált a (3.1) formula szerint számítjuk, ahol v_j már az új báziselem oszlopához tartozó régi potenciál lesz (i, j az új báziselem indexei); viszont, ha az új báziselem OS-elem, akkor az oszlopához tartozó új potenciált a (3.2) formula adja, melyben u_i a sorához tartozó régi potenciált jelenti. Ezután itt is az alapalgoritmus P4, P5 pontjait alkalmazzuk a következő változtatásokkal:

— a P4, P5 pontokat az új báziselemre nem vonatkoztatjuk,
— P5 pontból csak akkor térünk vissza a P4-hez, ha olyan SS-elem sorának potenciálja határozatlan, amelynek oszlopához már rendeltünk új potenciált, és amely nem új báziselem.

Végül maradnak olyan sorok és/vagy oszlopok, amelyekhez ez az algoritmus nem határoz meg potenciált, ezeken a helyeken a régi potenciálok maradnak érvényben.

A 4. és 5. fejezetben elmondottak bizonyíthatósága: A (4.1)—(4.6) összefüggések a sor-(oszlop-)út tulajdonságai és az (1.8) egyenlőségek alapján láthatók be, ezen felül a (4.3) formula nevezőjének nem-zéró voltát a bázisvektorrendszer lineáris függetlensége garantálja. Az 5. fejezet egyszerűsítései abból adódnak, hogy a báziselemek halmaza mind a sajátelemek kijelölése, mind a potenciálok meghatározása szempontjából általában particionálható. Közelebbről, például a potenciálok meghatározásával kapcsolatban, ez azt jelenti, hogy a potenciálokra a C1—C4 alapján felírható egyenletrendszer páronként idegen, valódi részrendszerekre bontható úgy, hogy különböző részrendszerhez tartozó két egyenlet nem tartalmaz közös potenciált.

6. A klasszikus feladat esete

Módosítsuk az (1.1) feladatot a

$$\begin{aligned} x_{m+1,j} &= 0, \quad j = 1, 2, \ldots, n; \\ x_{i,n+1} &= 0, \quad i = 1, 2, \ldots, m \end{aligned} \tag{6.1}$$

kikötésekkel. Így nem-normál lineáris programozási feladatot kaptunk, de mint az 1. fejezetben említettük, módszerünk kétfázisú változata erre is alkalmazható. Első fázis: (6.1) kielégítése; második fázis (6.1) megtartása mellett a megoldás további javítása.

Vegyük az (1.1) feladat (6.1) szerinti módosításának a következő megszorítását:

(6.2) $$f_{ij} = 1, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n;$$

(6.3) $$\sum_{i=1}^{m} t_i = \sum_{j=1}^{n} r_j.$$

Ez az ún. klasszikus szállítási feladat, és — mint speciális esetre — módszerünk természetesen rá is változtatás nélkül alkalmazható; azonban (6.1), (6.2), (6.3) további egyszerűsítéseket tesznek lehetővé. (6.1) és (6.3) miatt a fiktív elemektől eltekinthetünk, és nélkülük (6.2) miatt a feladat **A** mátrixának rangja $m+n-1$ lesz. — Így már valamely sorhoz vagy oszlophoz nem tudunk sajátelemet rendelni. Ezen körülménynek megfelelően itt átfogalmazzuk a klasszikus feladat esetére a sajátelem kijelölése, valamint az oszlop- és sorút kijelölése algoritmusát.

A sajátelemek kijelölése:

KS1. Minden sorban, amelyben még nincs SS-elem és csak egy kijelöletlen bázis-elemet tartalmaz, ezt a jelöletlen báziselemet jelöljük SS-elemnek.

KS2. Minden oszlopban, amelyben még nincs OS-elem, és csak egy jelöletlen báziselemet tartalmaz, ezt a báziselemet jelöljük OS-elemnek. Ha maradt még jelöletlen báziselem, akkor térjünk vissza a KS1 pontra.

Az oszlopút meghatározása:

KO1. Ha a bázisba bevonandó elem oszlopának nincs saját eleme (OS), akkor az oszlopút üres.

KO2. Egyébként elindulunk a bevonandó elemtől az oszlopában levő OS-elemhez; ha ennek sorában már nincs SS-elem, akkor ez az út vége, különben folytatjuk a KO3 pont szerint.

KO3. Az OS-elemből a sorában levő SS-elemhez lépünk; ha ennek oszlopában már nincs OS-elem, akkor ez az út vége, különben folytatjuk a KO4 pont szerint.

KO4. Az SS-elemből az oszlopában levő OS-elemhez lépünk; ha ennek sorában már nincs SS-elem, akkor ez az út vége, különben folytatjuk a KO3 pont szerint.

A sorút kijelölése hasonlóan történhet, ha az nem üres.

A feladat $m+n-1$ rangja, valamint a (6.2) megszorítás alapján a potenciálok számítására igaz:

— egy potenciál szabadon választható,

— a szabadon választott potenciál az összes többi potenciált meghatározza.

(6.2)-ből adódik az is, hogy a kompenzációs együtthatók hozzárendelése sor- és oszlopút esetén is így egyszerűsödik:

$$\begin{aligned} \mathbf{b}_1 &\to c_1 \quad = 1, \\ \mathbf{b}_2 &\to c_2 \quad = -1, \\ \mathbf{b}_{2k} &\to c_{2k} \quad = -1, \\ \mathbf{b}_{2k+1} &\to c_{2k+1} = 1. \end{aligned}$$

Könnyen belátható, hogy a klasszikus feladat esetében az ugyanazon (i, j)-hez tartozó sor- és oszlopút közös eleme(i) figyelmen kívül hagyható(k), mert az a_{ij} rá(juk) vonatkozó komponense(i) mindig zérus(ok).

IRODALOM

[1] GLOVER, F. and KLINGMAN, D., "Locating stepping-stone paths in distribution problems via the predecessor index method", *Transportation Science* 4 (1970) 220—225.
[2] GLOVER, F., KARNEY, D. and KLINGMAN, D., "The augmented predecessor index method for locating stepping-stone paths and assigning dual prices in distribution problems", *Transportation Science* 6 (1972) 171—180.
[3] KREKÓ, B., *Lineáris programozás* (Közgazdasági és Jogi Könyvkiadó, Budapest, 1966).
[4] POGÁNY, Zs., "An algorithm for solving the generalized transportation problem", *Egyetemi Számítóközpont Közleményei* 22 (1978) 61—74.
[5] VARGA, J., *Gyakorlati programozás* (Tankönyvkiadó, Budapest, 1972).

(Beérkezett: 1981. március 31.)

GYURKÓ GYÖRGY
PÉNZÜGYI ÉS SZÁMVITELI FŐISKOLA
3101 SALGÓTARJÁN, PF. 128.

AN ALGORITHM FOR THE SOLUTION OF THE GENERALIZED TRANSPORTATION PROBLEM BY THE AID OF THE CONCEPT OF LINE AND COLUMN EIGENELEMENTS

GY. GYURKÓ

This article presents a simple algorithm for solving the generalized transportation (distributional) problem. The algorithm is founded on the notions of the line- and column-eigenelements defined on the basic elements. In comparison to the *Glover's triple-label method* of the classical transportation problem ([3], [4]) this procedure can be regarded as an one-label method for the generalized problem.

Alkalmazott Matematikai Lapok 7 (1981) 217—252

SZTOCHASZTIKUS MÓDSZEREK OPTIMALIZÁLÁSI FELADATOK MEGOLDÁSÁRA

PINTÉR JÁNOS

Budapest

Dolgozatunkban a matematikai programozás nem-konvex és (vagy) nem-differenciálható szerkezetű feladatainak megoldására is alkalmas sztochasztikus optimalizálási eljárások vizsgálatával foglalkozunk, összefoglalva a témakörrel kapcsolatos korábbi eredményeinket is [48, 52]. A bevezetést követő 2. részben először nem feltétlenül konvex és differenciálható szerkezetű sztochasztikus programozási feladatokra vezető gyakorlati problémákat említünk, ezután a sztochasztikus feladatok megoldásának létezésével és egyértelműségével, valamint a korábbi optimalizálási eljárások konvergenciájával kapcsolatos eredményeket foglaljuk össze. A 3. részben először a sztochasztikus optimalizálási módszerek néhány alapvető konvergenciatulajdonságát igazoljuk; a továbbiakban ezeket az eredményeket terjesztjük ki sztochasztikusan kombinált (hibrid) módszerek konvergenciájának vizsgálatára. A dolgozat 4. részében a hibrid módszerek gépi realizációival elvégzett néhány számítógépes kísérlet eredményeit ismertetjük.

1. Bevezetés

A műszaki, gazdasági rendszerek tervezésének és operatív irányításának feladatai gyakran korlátozó feltételek melletti optimalizálási probléma alakjában fogalmazhatók meg. Ha a döntési alternatívák az E^n n-dimenziós euklideszi tér pontjaival reprezentálhatók, akkor az optimalizálási feladatok egy általános típusa a következőképpen írható fel:

$$\min f(\mathbf{x}) \tag{1.1}$$

$$g_i(\mathbf{x}) \geqq 0, \quad i = 1, \ldots, m.$$

Itt az $f: E^n \to E^1$ célfüggvény és a $g_i: E^n \to E^1$ korlátozó feltételek az $\mathbf{x}$ döntési változó függvényei. Az (1.1) típusú feladatok szerkezetének és megoldási módszereinek vizsgálatával a matematikai programozás foglalkozik. Bár a matematikai programozás csak néhány évtizedes múlttal rendelkezik, a témakör irodalma ma már igen kiterjedt (példaként a [66, 76, 83] munkákat és a bennük szereplő további hivatkozásokat említjük).

A matematikai programozási feladatok túlnyomó többségét a feladat célfüggvényére és korlátozó feltételeire vonatkozó tökéletesen pontos információ feltételezésével, determinisztikus alakban fogalmazzák meg. Nyilvánvaló ugyanakkor, hogy a döntés számos esetben véletlen tényezőktől is függő körülmények között kerül kiválasztásra. Ezért nagy jelentőséggel bír a véletlentől függő döntési feladatok helyes megfogalmazása és az ilyen feladatok megoldására szolgáló eljárások létrehozása; a sztochasztikus (matematikai) programozás tárgyát az ezzel kapcsolatos vizsgálatok képezik [27, 53, 55, 80, 96, 97].

Egy matematikai programozási feladatnak konkrét tartalmától függően többféle sztochasztikus kiterjesztése lehetséges. Pl. az (1.1) feladat egy sztochasztikus variánsa az alábbi alakú lehet:

$$\min Mf(\mathbf{x}, \mathbf{y}) \tag{1.2}$$

$$Mg_i(\mathbf{x}, \mathbf{y}) \geqq 0, \quad i = 1, \ldots, m.$$

Az (1.2) feladatban szereplő $\mathbf{y}$ a célfüggvény és a korlátozó feltételek értékeit befolyásoló véletlen tényezőket jelöli (a legtöbb esetben feltesszük, hogy $\mathbf{y}$ E^q-beli értékkészletű valószínűségi vektorváltozó), az $f, g_i: E^{n+q} \to E^1$ függvények az (1.1) feladat függvényeinek felelnek meg, M pedig a várható érték képzésének operátora.

Az (1.2) probléma vizsgálata során alapvető fontossággal bír az a körülmény, hogy a feladat célfüggvényének és korlátozó feltételeinek értékei általában analitikusan nem számíthatók ki, csak valamilyen véletlen hibával becsülhetők, a becslés pontosságának növelése pedig a felhasznált erőforrások (kísérletek, számítógépi szimuláció) mennyiségének növelését vonja maga után. Ezért sztochasztikus optimalizálási feladatok megoldása során olyan eljárásokra van szükség, amelyek a közelítő megoldások sorozatának előállítási módját és a megfelelő függvénykiértékelések pontosságát egyaránt meghatározó stratégiát szolgáltatnak.

A matematikai programozási feladatok megoldására irányuló korábbi iteratív sztochasztikus módszerek (pl. [7, 17, 29, 34, 48, 67, 73, 80, 86, 90, 93]) lényegileg két csoportra oszthatók: véletlen kereső, ill. sztochasztikus approximáció típusú eljárásokra. A véletlen kereső módszereket gyakran heurisztikus stratégiaként, többnyire determinisztikus feladatok megoldására alkalmazták. Ezek elméleti konvergenciája enyhe feltételek mellett biztosítható, numerikus hatékonyságukkal kapcsolatban viszont eléggé eltérő vélemények alakultak ki. A sztochasztikus approximáció jellegű módszereket, ill. azok bizonyos (sztochasztikus kvázigradiens) típusú kiterjesztéseit viszont általában csak feltétel nélküli, ill. konvex struktúrájú feladatok megoldására alkalmazták, a feladat függvényeinek differenciálhatóságára (ill. a sztochasztikus kvázigradiensek halmazára) vonatkozó különböző feltevések mellett. Ezek a feltételek a gyakorlatban nem mindig teljesülnek, ill. ellenőrzésük nehézségekbe ütközhet (ezzel kapcsolatban l. még pl. [80] I. fejezet 3—4. §).

Dolgozatunkban a matematikai programozás nem-konvex és (vagy) nem-differenciálható szerkezetű feladatainak megoldására is alkalmas sztochasztikus módszerek vizsgálatával foglalkozunk, összefoglalva a témakörrel kapcsolatos korábbi eredményeinket (l. pl. [48, 52]) is. A bevezetést követő 2. részben nem feltétlenül konvex és differenciálható szerkezetű sztochasztikus programozási feladatokra vezető gyakorlati problémákat említünk, a sztochasztikus feladatok megoldásának létezésével és egyértelműségével kapcsolatos eredményeket, és a korábbi iteratív optimalizálási eljárások konvergenciatulajdonságait foglaljuk össze. A 3. részben először a sztochasztikus optimalizálási eljárások néhány alapvető konvergenciatulajdonságát igazoljuk; a továbbiakban ezeket az eredményeket terjesztjük ki sztochasztikusan kombinált (hibrid) módszerek konvergenciájának vizsgálatára. A dolgozat 4. részében a hibrid módszerek gépi realizációival elvégzett néhány számítógépes kísérlet eredményeit ismertetjük.

Dolgozatunkban — saját eredményeink összefoglaló ismertetésén túlmenően — a sztochasztikus optimalizálási eljárásokra vonatkozó áttekintést is kívánunk nyújtani. Ezzel kapcsolatban megjegyezzük, hogy az irodalomjegyzék ésszerű rövidítése

érdekében egy-egy témakörrel kapcsolatban gyakran csak összefoglaló jellegű munkákat, cikkgyűjteményeket említünk; emellett a hazai szerzők eredményeit ismertető publikációk közül lehetőség szerint a magyar nyelvűekre hivatkozunk.

A sztochasztikus programozás iránti érdeklődésemet elsősorban PRÉKOPA ANDRÁS munkássága keltette fel: az általa és munkatársai által folytatott kutatások munkámra mindvégig ösztönzően hatottak. Köszönettel tartozom JU. M. JERMOLJEVnek, B. T. POLJAKnak és A. G. ZILINSKASnak a munkámmal kapcsolatos és a témakör szovjet szakirodalmában való tájékozódásomat segítő hasznos tanácsokért, KOVÁCS MARGITnak és UGRAY LÁSZLÓNÉnak pedig a kézirat átnézésével kapcsolatos megjegyzéseikért.

2. Sztochasztikus programozási feladatok; strukturális tulajdonságok és megoldási módszerek

Ebben a fejezetben először néhány olyan gyakorlati feladatot említünk, amelyek sztochasztikus programozási problémaként fogalmazhatók meg. Ezt követően a sztochasztikus modellek szerkezetére vonatkozó néhány általános eredményt ismertetünk és a megoldásukra szolgáló korábbi iteratív eljárások tulajdonságait foglaljuk össze.

2.1. *Néhány sztochasztikus optimalizálási feladat*

Vízgazdálkodási modellek

Földünk népességének rohamos növekedése és a mind bonyolultabbá váló gazdasági feltételek következtében a természetes erőforrások felhasználásának észszerű tervezése egyre fokozódó jelentőséggel bír. Ez a tendencia a vizgazdálkodási rendszerek tervezésével kapcsolatos kutatások kiterjesztését is eredményezte. A tervezés folyamán egyrészt figyelembe kell venni a változó természeti körülményeket, másrészt a döntési alternatívák összetett műszaki-gazdasági jellemzőit és egyéb (jogi, szociális, politikai stb.) szempontokat is. A vízgazdálkodási rendszerek általában nyilvánvaló módon illusztrálják azt a körülményt, hogy a döntéseket különböző véletlen tényezők lényegesen befolyásolják. Példaként sztochasztikus vízminőségszabályozási és tározórendszer tervezési modelleket említünk meg és csupán utalunk a témakör gazdag irodalmára (l. pl. a [16, 25, 35, 51, 88] munkákat és az ott szereplő hivatkozásokat).

Vízminőségszabályozás

Tekintsünk egy vízrendszert, amelynek folyóit különböző pontokban szennyezések terhelik. Feltesszük, hogy a (néhány vízminőségi paraméterrel mért) szennyezőhatások olyan jelentősek, hogy a kibocsátási pontokban tisztítási technológiák alkalmazása szükséges ahhoz, hogy a víz előírt minőségét a vizsgált térségben fenntarthassuk. A vízminőségi előírások teljesülését adott mérési pontokban ellenőrizzük: ezek nem szükségképpen azonosak a kibocsátási pontokkal, tehát a vízrendszer öntisztulási képességével is számolnunk kell. A feladat az adott térség vízminőség-

szabályozási költségeinek minimalizálása. (Megjegyezzük, hogy ez a probléma általánosabb keretek között is tárgyalható (l. pl. [16, 43, 44, 51]). Megoldására korábban gyakran fogalmaztak meg statikus és determinisztikus jellegű, általában lineáris, konvex vagy diszkrét programozási problémára vezető modelleket [35]. [51] dolgozatunkban egy olyan sztochasztikus programozási modellt ismertettünk, amely a feladatban szereplő véletlen tényezőket (hidrológiai feltételek, szennyvíztelepek hatásfoka stb.) is figyelembevéve a vízminőségszabályozás adott időszakra vonatkozó összegzett költségeinek várható értékét minimalizálja, a vízminőségi előírások betartásának valószínűségére vonatkozó korlátozó feltétel és további determinisztikus (műszaki jellegű) feltételek mellett.

Többcélú tározórendszer tervezése

Különböző ipari, mezőgazdasági, kommunális vízellátási, vízminőségszabályozási, árvízvédelmi stb. követelmények kielégítésének egyik lehetséges módja víztározók segítségével valósítható meg. A tározórendszerek tervezésével kapcsolatban nagyszámú modell ismeretes (l. pl. [1, 4, 16, 41, 57, 60, 61, 62, 64]). [44, 46] munkáinkban a *Sajó-térség* vízminőségszabályozási programjával kapcsolatban a tározókkal történő vízhozam-kiegyenlítés lehetőségét elemeztük és kapcsoltuk össze egy vízminőségvédelmi beruházási modellel. [50] dolgozatunkban a *Bódva völgyében* megépíthető tározórendszerre vonatkozó döntési alternatívákat vizsgáltuk. Az említett modelleket — figyelembe véve a hidrológiai tényezők véletlen jellegét — sztochasztikus alakban fogalmaztuk meg. A feladat a tározórendszer építési költségeinek és a ki nem elégített vízigényekből származó várható veszteségek összegének minimalizálása volt, a vízigények kielégítésére vonatkozó valószínűségi feltételek és további műszaki korlátozások mellett.

Készletmodellek

A gazdasági életben jelentős szerepe van a termelő egységek folyamatos anyagellátásának. Az ezzel kapcsolatos készletgazdálkodási feladatok vizsgálatára számos matematikai modellt dolgoztak ki (l. pl. [3, 28, 57]). Ezek közül itt csak az elsőként hazai szerzők [54, 71] által vizsgált megbízhatósági típusú készletmodelleket, ill. azok további változatait említjük meg: ezeket a [28] tanulmány foglalja össze. Az egy termék (anyag, árucikk) vizsgálatára szorítkozó modellek általános jellemzője, hogy — a sztochasztikus igény-, ill. beérkezési folyamatok szerkezetére vonatkozó különböző feltevések mellett — a vizsgált időperiódus kezdőkészlete viszonylag egyszerű egzakt vagy aszimptotikus formula segítségével számítható. Ilyen jellegű eredményt igazoltunk pl. [45] dolgozatunkban, a [47]-ben pedig ezt egy numerikus példával is illusztráltuk. (Analóg módon vizsgálható — adott kezdőkészletet feltételezve — a raktárkapacitás megbízhatósági szintű tervezésének feladata is). A többtermékes esetben a feladat általánosabb jellegű sztochasztikus programozási problémára vezet.

A fenti gyakorlati példákon túlmenően a sztochasztikus programozásnak még számos alkalmazási lehetősége ismert (l. pl. a [26, 27, 55, 80, 97] munkákban vizsgált modelleket).

2.2. *Sztochasztikus programozási feladatok minőségi tulajdonságai*

Az előző szakaszban különböző gyakorlati problémák sztochasztikus modelljeire vonatkozó példákat említettünk; a továbbiakban a sztochasztikus programozási feladatok néhány általános szerkezeti tulajdonságát vizsgáljuk meg. A feladatok elvi megoldhatósága szempontjából alapvetőek az optimális megoldás létezésének elégséges feltételei. Ezzel kapcsolatban WEIERSTRASS közismert tétele (l. pl. [84], 101—103. o.) alkalmazható.

Tekintsük először a gyakran vizsgált, alábbi típusú sztochasztikus programozási feladatot (l. pl. [55—57]):

$$\min Mf(\mathbf{x}, \mathbf{y}) \tag{2.1}$$

$$P(g_i(\mathbf{x}, \mathbf{y}) \geqq 0,\ i = 1, \ldots, m) \geqq p$$

$$\mathbf{x} \in K \subset E^n.$$

Itt $\mathbf{x} \in E^n$ döntési változó, az n-dimenziós euklideszi tér pontja; $\mathbf{y}$ vektorértékű valószínűségi változó, vagyis egy $(\Omega, \mathcal{A}, P)$ valószínűségi mező Ω alaphalmazát az E^q euklideszi térbe képező P-mérhető függvény; $f, g_i: E^{n+q} \to E^1$ *Borel-mérhető* valós függvények; M a várható érték operátora; $P(\cdot)$ a $(\cdot)$ véletlen esemény bekövetkezésének valószínűsége; $0 < p < 1$ ennek a valószínűségnek előírt alsó korlátja; K pedig az $\mathbf{x}$ vektorra vonatkozó egyéb determinisztikus feltételek által meghatározott halmaz. Itt és a továbbiakban feltesszük, hogy a képletekben szereplő várható értékek léteznek.

Tekintsük még a következő feladattípusokat (l. pl. [58, 80, 97]):

$$\min Mf(\mathbf{x}, \mathbf{y}) \tag{2.2}$$

$$Mg_i(\mathbf{x}, \mathbf{y}) \geqq 0, \quad i = 1, \ldots, m$$

$$\mathbf{x} \in K;$$

$$\min Mf(\mathbf{x}, \mathbf{y}) \tag{2.3}$$

$$M(y_i - g_i(\mathbf{x}) | y_i - g_i(\mathbf{x}) > 0) \leqq d_i, \quad i = 1, \ldots, m$$

$$\mathbf{x} \in K \quad \mathbf{y} = (y_1, \ldots, y_m).$$

A (2.2) feladatban a $g_i(\mathbf{x}, \mathbf{y}) \geqq 0$ (véletlentől függő) feltételek várható értékben való teljesülését írjuk elő. A (2.3) feladatban az $\mathbf{y}$ valószínűségi vektorváltozó komponenseire vonatkozó $y_i - g_i(\mathbf{x}) \leqq 0$, $i = 1, \ldots, m$ (véletlentől függő) előírásoktól való eltérés feltételes várható értékét korlátozzuk; itt $g_i: E^n \to E^1$, $i = 1, \ldots, m$ mérhető valós függvények, a d_i, $i = 1, \ldots, m$ értékek pedig megadott pozitív állandók.

Nem nehéz megmutatni, hogy a (2.1)—(2.3) feladatoknak — lényegileg csak a megfelelő f, g_i függvények $\mathbf{x}$ szerinti folytonosságát feltételezve (a szereplő valószínűségi változók majdnem minden realizációja mellett) — létezik optimális megoldása, függetlenül a valószínűségi változók típusától. A bizonyítás *Weierstrass említett tételén* (vagyis a megengedett tartomány kompaktságának és a célfüggvény

folytonosságának igazolásán) alapszik. (L. pl. a (2.1) feladattal kapcsolatban [52] 5.1 tételét, amely az ott leírtaknál valamivel enyhébb feltevések mellett is igazolható.) Természetesen a (2.1)—(2.3) modellek feltételtípusai egy modellen belül is szerepelhetnek. Megemlítjük még, hogy sztochasztikus programozási feladatok optimális megoldásának létezésével foglalkoznak pl. a [89, 91, 92] dolgozatok, valamint [97] II. fejezetének 1. és 6. pontja.

Érdemes itt megjegyezni, hogy a sztochasztikus programozási feladatok differenciálható struktúrája csak erősebb feltételek mellett biztosítható. Pl. az $Mf(\mathbf{x}, \mathbf{y})$ célfüggvény $\mathbf{x}$ szerinti differenciálhatóságához szükséges az f függvény $\mathbf{x}$ szerinti differenciálhatósága ($\mathbf{y}$ majdnem minden realizációja esetén). Ez a feltétel nem áll fenn pl. ha a 2.1 szakaszban említett optimalizálási feladatok célfüggvényeiben a nem teljesülő valószínűségi korlátozásokból adódó, nem-differenciálható függvényekkel képezett büntetőtagok is szerepelnek.

A sztochasztikus programozási feladatok optimális megoldásának létezésére vonatkozó feltételek önmagukban nem elegendőek a megoldás számítási szempontból is hatékony meghatározásához. Egyrészt a vizsgált feladatnak több lokális optimumhelye is lehet, másrészt a szereplő függvények kiértékelése általában csak időigényes és véletlen hibákat eredményező becslési módszerekkel végezhető el. Az említett problémák közül most csak a lokális, ill. globális optimumhelyek kérdését elemezzük, a zajhatásokkal torzított függvényeken alapuló optimalizálási módszereket később vizsgáljuk.

Ismeretes, hogy a nemlineáris programozási módszerek döntő többsége csak lokális optimumhelyek meghatározását teszi lehetővé. Ezért a globális optimumhely kiválasztása még determinisztikus feladat esetében is nehéz problémát jelenthet. Csak néhány speciális szerkezetű feladattípus (pl. a konvex programozási probléma) rendelkezik azzal a tulajdonsággal, hogy lokális optimumhelyei egyben globális megoldások is. Ezért igen jelentős azoknak a feltételeknek a vizsgálata, amelyek a sztochasztikus programozási feladatok említett struktúráját biztosítják.

Az első valószínűséggel korlátozott modelleket egyszerűen meghatározható, konvex, determinisztikus ekvivalensre vezető alakban fogalmazták meg (l. pl. [11, 12, 40, 42, 70]). Ez természetesen csak a sztochasztikus modellek eléggé szűk osztálya esetében lehetséges (a főként az ilyen modellekre érvényes kritikai megjegyzéseket a [23] dolgozat foglalja össze). Jóval általánosabb sztochasztikus programozási modellek konvex struktúráját biztosítják az [56, 57] dolgozatok eredményei (itt a feladat determinisztikus ekvivalens alakja általában már nem ismert). Ezzel kapcsolatban csak egy általános eredményt idézünk.

2.1. TÉTEL. ([57], 2. tétel). Tekintsük (a korábban bevezetett jelölésekkel) a

$$G(\mathbf{x}) = P(g_i(\mathbf{x}, \mathbf{y}) \geqq 0,\ i = 1, \ldots, m) \geqq p \tag{2.4}$$

típusú valószínűségi feltételt. Ha az $\mathbf{y}$ valószínűségi vektorváltozó logaritmikusan konkáv sűrűségfüggvénnyel rendelkezik és a g_i, $i=1, \ldots, m$ függvények konkávak, akkor G logaritmikusan konkáv függvénye $\mathbf{x}$-nek.

Ebből a tételből következik, hogy ha egy optimalizálási feladat egyéb feltételei által definiált megengedett tartomány konvex, akkor ez teljesül a (2.4) típusú feltétellel kiegészített feladatra is. A 2.1. tételnek számos alkalmazása ismert, pl. [57—62]. A korábbi jelölések felhasználásával tekintsük pl. a következő feladatot (vö. az [58]

dolgozatban szereplő (2.1) modellel):

$$\min\left\{f(\mathbf{x})+\sum_{i=1}^{m} Q_i\Big(\int_{g_i(\mathbf{x})}^{\infty}[z-g_i(\mathbf{x})]\,dH_i(z)\Big)\right\} \tag{2.5}$$

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

$$M(y_i-g_i(\mathbf{x})\,|\,y_i-g_i(\mathbf{x})>0) \leqq d_i,\quad i=1,\ldots,m$$

$$\mathbf{x}\in K\subset E^n.$$

A 2.1. tétel alapján egyszerűen igazolható, hogy ha (2.5) megengedett megoldásainak halmaza nem üres, a $-f, g_i$, $i=1,\ldots,m$ függvények konkávak a konvex, kompakt K halmazon, az y_i, $i=1,\ldots,m$ valószínűségi változók $H_i(z)$ eloszlásfüggvényei abszolút folytonosak, sűrűségfüggvényeik és együttes sűrűségfüggvényük logaritmikusan konkávak, végül a $Q_i(w_i)$, $i=1,\ldots,m$ büntetőfüggvények

$$w_i = w_i(\mathbf{x}) = \int_{g_i(\mathbf{x})}^{\infty}[z-g_i(\mathbf{x})]\,dH_i(z)$$

argumentumaiknak monoton növekvő konvex függvényei, akkor (2.5) konvex programozási feladat.

Megjegyezzük, hogy bár a 2.1. tétel a sztochasztikus programozási feladatok eléggé széles osztályával kapcsolatban alkalmazható, a konvex szerkezet a sztochasztikus feladatoknak nem általános tulajdonsága. Nyilvánvaló pl., hogy ha az $F(\mathbf{x})=$ $=Mf(\mathbf{x},\mathbf{y})$ célfüggvényben $(\mathbf{x}\in E^n,\ \mathbf{z}=\mathbf{y}(\omega)\in E^q)$, $f: E^{n+q}\to E^1$ mérhető függvénye $\mathbf{z}$-nek és 1 valószínűséggel konkáv függvénye $\mathbf{x}$-nek, akkor $F(\mathbf{x})$ tetszőleges $K\subset E^n$ konvex halmazon konkáv függvény. Ezért a $\min_{\mathbf{x}\in K} F(\mathbf{x})$ feladat már rendelkezhet több lokális minimummal, amelyek közül a globális megoldás kiválasztása nehézségekbe ütközhet. (Igazolható pl., hogy ha $F(\mathbf{x})$ (kvázi)konkáv a K kompakt halmaz konvex burkán, akkor van olyan globális optimumhely, amelyik K-nak határpontja.) Természetesen összetettebb példák is mutathatók nem-konvex sztochasztikus programozási feladatokra, pl. lehetséges, hogy a megengedett tartomány nem konvex (l. a 2.1. tétel feltételeit). Ezekben az esetekben — ha a modell konvex átfogalmazása a vizsgált feladat lényeges elemeinek mellőzése nélkül nem lehetséges — olyan optimalizálási eljárást kell választanunk, amely a konvexitási feltételek nélkül is alkalmazható az optimális megoldás meghatározására. Mielőtt egy ilyen módszertípust részletesebben megvizsgálnánk, röviden áttekintjük a sztochasztikus programozási feladatok megoldására korábban alkalmazott módszereket.

2.3. *Iteratív eljárások konvex és differenciálható struktúrájú sztochasztikus programozási feladatok megoldására*

Iteratív sztochasztikus algoritmusok általános struktúráját és konvergenciatulajdonságait pl. a [7, 17, 29, 34, 53, 73, 80, 90, 96, 97] munkákban vizsgálták; az alábbi 2.2., 2.3., illetve 2.4. konvergenciatételeket a [90] dolgozat, illetve a [17, 80] munkák alapján ismertetjük. Az algoritmus sztochasztikus jellege a probléma saját

strukúrájából és(vagy) az optimalizálási módszer mesterséges sztochasztikussá tételéből adódhat: az alábbi tárgyalásmód ezek egységes kezelését teszi lehetővé.

Tegyük fel, hogy valamely $X^* \subset E^n$ halmaz elemeinek meghatározására az

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma_k \mathbf{s}_k, \quad k = 0, 1, 2, \ldots \tag{2.6}$$

iterációt alkalmazzuk. Itt k az iteráció sorszáma, $\mathbf{x}_k \in E^n$ a megoldás k-adik közelítése, $\mathbf{s}_k \in E^n$ a k-adik lépés iránya, $\gamma_k \in E^1$ pedig a lépés hossza. Tegyük fel, hogy (2.6)-ban $\mathbf{s}_k$, és γ_k kiválasztása a következő feltételek szerint történik (a [90]-ben szereplő A.—E. feltételek speciális alakjának ismertetésére szorítkozunk):

1. Az $\mathbf{s}_k$ vektor eloszlása csak $\mathbf{x}_k$-tól függ, az $\mathbf{s}_k$-k egymástól függetlenek és véletlen összetevőjük 0 várható értékű:

$$\mathbf{s}_k = \mathbf{s}_k(\mathbf{x}_k), \quad \mathbf{s}_k(\mathbf{x}_k) = \mathbf{t}_k(\mathbf{x}_k) + \boldsymbol{\xi}_k(\mathbf{x}_k), \quad M\boldsymbol{\xi}_k(\mathbf{x}_k) = \mathbf{0}.$$

2. A keresett X^* halmaznak megfelelően adott egy $V(\mathbf{x})$ függvény, amely kielégíti a $V(\mathbf{x}) \geqq 0$, $\inf_{\mathbf{x} \in E^n} V(\mathbf{x}) = 0$ feltételeket, emellett differenciálható és gradiense az E^n-ben egyenletes *Lipschitz-feltételnek* tesz eleget:

$$\|\nabla V(\mathbf{z}_1) - \nabla V(\mathbf{z}_2)\| \leqq L\|\mathbf{z}_1 - \mathbf{z}_2\|. \quad (L > 0 \text{ állandó})$$

3. Az algoritmus „átlagosan a $V(\mathbf{x})$ csökkenése irányába halad", ezt a következő relációval írjuk elő:

$$(\nabla V(\mathbf{x}_k), \mathbf{t}_k(\mathbf{x}_k)) \leqq -\nu V(\mathbf{x}_k). \quad (\nu > 0 \text{ állandó}).$$

4. Az irányvektor hosszának átlagos nagyságrendjét a

$$\|\mathbf{t}_k(\mathbf{x}_k)\|^2 + M\|\boldsymbol{\xi}_k(\mathbf{x}_k)\|^2 \leqq \sigma^2 + \tau V(\mathbf{x}_k) \quad (\sigma \geqq 0, \tau > 0 \text{ állandó})$$

egyenlőtlenséggel korlátozzuk.

5. Az iteráció $\mathbf{x}_0$ kezdőpontjában fennáll

$$MV(\mathbf{x}_0) < \infty.$$

6. A bevezetett eljárás-paraméterek függvényében a lépéshosszak kielégítik az alábbi relációkat:

$$0 \leqq \gamma_k \leqq \frac{2(\nu - \varepsilon)}{L} \quad (0 < \varepsilon < \nu); \quad \sum_{k=0}^{\infty} \gamma_k = \infty.$$

Az 1.—6. feltételek fennállása esetén a (2.6) iteratív sztochasztikus algoritmus alábbi alapvető konvergenciatulajdonságai igazolhatók ([90] 2., 3. tétel):

2.2. TÉTEL. Ha $\lim_{k \to \infty} \gamma_k = 0$, akkor $\lim_{k \to \infty} MV(\mathbf{x}_k) = 0$.

2.3. TÉTEL. Ha $\sigma = 0$ vagy $\sum_{k=0}^{\infty} \gamma_k^2 < \infty$, akkor $P\left(\lim_{k \to \infty} V(\mathbf{x}_k) = 0\right) = 1$.

Ezekből a tételekből — a V indikátorfüggvényre megfelelő feltételeket szabva — az iterációnak az X^* halmazhoz négyzetes középben, ill. 1 valószínűséggel való konvergenciája adódik. Az említett további munkák hasonló jellegű konvergencia-eredményeket tartalmaznak.

Mielőtt a fenti általános eredményeknek a sztochasztikus programozásban való alkalmazására áttérnénk, vizsgáljuk meg az 1.—6. feltételeket. Nem nehéz végig-

gondolni, hogy az 1., 4. és 5. feltételek viszonylag egyszerűen kielégíthetők, illetve jól közelíthetők. Problémát okozhat viszont a megfelelő $V(\mathbf{x})$ függvény kiválasztása (2.), illetve az ennek megfelelő haladási irányok meghatározása (3.). Érdemes még rámutatni arra is, hogy a szereplő $L, \nu, \sigma, \tau, \varepsilon$ állandók a gyakorlatban nem ismertek és becslésük sem egyszerű feladat (6.).

A fenti általános konvergenciatételeknek matematikai programozási feladatokra való alkalmazása például az optimalitás elégséges, ill. szükséges feltételeit kimondó nyeregpont-egyenlőtlenség, ill. a *Kuhn—Tucker-tétel* alapján történhet. A matematikai programozás említett általános eredményeit pl. [83] 3. fejezete vagy [66] 6. fejezete részletezi, ezért itt csak a sztochasztikus feladat esetére vonatkozó tényeket ismertetjük. Tegyük fel, hogy megoldandó a

$$\min Mf(\mathbf{x}, \mathbf{y}) \tag{2.7}$$

$$Mg_i(\mathbf{x}, \mathbf{y}) \leqq 0, \quad i = 1, \ldots, m$$

sztochasztikus programozási probléma ($\mathbf{x} \in E^n$ döntési változó, $\mathbf{y}$ a feladat véletlen tényezőit reprezentáló E^q-beli értékkészletű valószínűségi vektorváltozó, $f, g_i: E^{n+q} \to E^1$ mérhető valós függvények). Tekintsük a (2.7) feladatnak megfelelő Lagrange-függvényt:

$$L(\mathbf{x}, \mathbf{y}, \mathbf{u}) = Mf(\mathbf{x}, \mathbf{y}) + \sum_{i=1}^{m} u_i Mg_i(\mathbf{x}, \mathbf{y})$$

$$(u_i \geqq 0,\ i = 1, \ldots, m, \text{ azaz } \mathbf{u} = (u_1, \ldots, u_m) \in E_+^m).$$

Az említett általános eredményekből adódik, hogy ha az $L(\mathbf{x}, \mathbf{y}, \mathbf{u})$ függvénynek létezik nyeregpontja (ez egyben (2.7) optimális megoldását is megadja), akkor ez elvileg meghatározható a

$$\min_{\mathbf{x} \in E^n} \max_{\mathbf{u} \in E_+^m} L(\mathbf{x}, \mathbf{y}, \mathbf{u}), \quad \max_{\mathbf{u} \in E_+^m} \min_{\mathbf{x} \in E^n} L(\mathbf{x}, \mathbf{y}, \mathbf{u}) \tag{2.8}$$

feladatok bármelyikének megoldásával. Másrészt a 2.2.—2.3. tételek 2. feltételében szereplő $V(\mathbf{x})$ indikátorfüggvény a *Kuhn—Tucker-tétel* alapján a következőképpen definiálható:

$$V(\mathbf{x}) = \left\| \nabla_{\mathbf{x}} Mf(\mathbf{x}, \mathbf{y}) + \sum_{i=1}^{m} u_i^* \nabla_{\mathbf{x}} Mg_i(\mathbf{x}, \mathbf{y}) \right\|^2. \tag{2.9}$$

Itt u_i^*, $i = 1, \ldots, m$ a (2.7) feladat optimális megoldásához a *Kuhn—Tucker-tétel* szerint hozzárendelt *Lagrange-szorzók*. A (2.8) típusú feladatok iteratív megoldása — még determinisztikus függvényekkel is — csak speciális esetekben hajtható végre, a (2.9)-ben szereplő $V(\mathbf{x})$ függvény differenciálhatóságához pedig a (2.7) feladat f, g_i függvényeinek $\mathbf{x}$ szerinti kétszeri differenciálhatósága szükséges. Ezért az iteratív algoritmusoknak sztochasztikus feladatokra való alkalmazása során általában feltételezik a feladat konvex és differenciálható struktúráját. Emellett maguk a módszerek gradiens típusú eljárásokon, illetve ezeknek egyszerű véletlen kereső technikákkal való kiegészítésein alapulnak. Példaként a *Lagrange-* vagy a *Hestenes—Powell multiplikátor-módszer* [19, 33, 53], a büntetőfüggvény-módszer [53, 60, 78], a megengedett irányok módszere [31, 32, 59, 74] a redukált gradiens módszer [37], VEINOTT támaszsík módszere [68], vagy a gradiens-projekciós módszer [36] alkalmazásait említjük meg.

Bizonyos mértékben általánosabb konvergenciaeredmények igazolhatók sztochasztikus kvázigradiens típusú módszerekre (részletes vizsgálatuk pl. a [80] munkában található meg, [17] a szóban forgó módszerek rövid áttekintését tartalmazza). Tekintsük először a

$$\min_{\mathbf{x}\in X} f(\mathbf{x}) \tag{2.10}$$

feladatot, ahol f konvex (folytonos) függvény az $X\subset E^n$ konvex, korlátos és zárt halmazon. Legyen $X^*\subset X$, $X^*\neq\emptyset$ az optimumhelyek halmaza. A (2.10) feladat megoldását a tetszőleges $x_0\in X\setminus X^*$ pontból indított

$$\mathbf{x}_{k+1} = \Pi_X(\mathbf{x}_k-\gamma_k\mathbf{s}_k), \quad k=0, 1, 2, \ldots \tag{2.11}$$

iterációval végezzük. Itt Π_X — az X halmazra vonatkozó projekció operátora, $\gamma_k\geqq 0$ és $\mathbf{s}_k\in E^n$ pedig valószínűségi változók: γ_k az iterációs lépés hossza, $\mathbf{s}_k$ az f függvény sztochasztikus kvázigradiense, amely $\mathbf{x}\in X$ esetén az

$$f(\mathbf{x})-f(\mathbf{x}_k) \geqq (\{M\mathbf{s}_k|\mathbf{x}_0, \ldots, \mathbf{x}_k\}, \mathbf{x}-\mathbf{x}_k)+\mathbf{b}_k(\mathbf{x}_0, \ldots, \mathbf{x}_k) \tag{2.12}$$

relációnak tesz eleget. A (2.12) egyenlőtlenségben szereplő $\mathbf{b}_k\in E^n$ általános esetben az $\mathbf{x}_0, \ldots, \mathbf{x}_k$ pontoktól függő valószínűségi vektorváltozó realizációja. Megjegyezzük, hogy ha $M\mathbf{b}_k\equiv 0$, $k=0, 1, 2, \ldots$, akkor (2.12) értelmében az $\mathbf{s}_k$ irány várható értéke az $f(\mathbf{x})$ függvény általánosított gradiense.

A (2.11) iterációval kapcsolatban érvényes a következő állítás (vö. [17] 2. tételével; általánosabb konvergenciaeredményeket [80] III.—IV. fejezetei tartalmaznak).

2.4. TÉTEL. Tegyük fel, hogy fennállnak a következő relációk:

$$M(\|\mathbf{s}_k\|^2|\mathbf{x}_0, \ldots, \mathbf{x}_k) \leqq C, \quad k=0, 1, 2, \ldots \quad (C>0 \text{ állandó}),$$

$$\sum_{k=0}^{\infty}\gamma_k=\infty, \quad \sum_{k=0}^{\infty}M\gamma_k\|\mathbf{b}_k\|<\infty, \quad \sum_{k=0}^{\infty}M\gamma_k^2<\infty \quad (1 \text{ valószínűséggel}).$$

Ekkor az $\{\mathbf{x}_k\}$ sorozat minden határpontja 1 valószínűséggel az X^* halmazhoz tartozik.

Megemlítjük még, hogy a fenti tételhez hasonló konvergenciaeredmény a (2.10) feladattal kapcsolatban akkor is alkalmazható, ha az f függvény az X halmazon folytonos és gyengén konvex, l. [80] V. fejezet 1.—2. §. A gyenge konvexitás azt jelenti, hogy minden $\mathbf{x}\in X$ esetén létezik olyan $G(\mathbf{x})\subset E^n$ $G(\mathbf{x})\neq\emptyset$ vektorhalmaz, hogy minden $\hat{f}_{\mathbf{x}}\in G(\mathbf{x})$ és $\mathbf{z}\in X$ esetén fennáll

$$f(\mathbf{z})-f(\mathbf{x}) \geqq (\hat{f}_{\mathbf{x}}, \mathbf{z}-\mathbf{x})+r(\mathbf{z}, \mathbf{x}), \quad \text{ahol} \quad r(\mathbf{z}, \mathbf{x}) = o(\|\mathbf{z}-\mathbf{x}\|).$$

(Nyilvánvaló, hogy a konvex vagy a differenciálható függvények gyengén konvexek, az $r(\mathbf{z}, \mathbf{x})\equiv 0$ esetben pedig $\hat{f}_{\mathbf{x}}$ az f függvény általánosított gradiense.)

3. Sztochasztikus módszerek optimalizálási feladatok megoldására

A 2.1.—2.2. szakaszokban konkrét feladattípusok bemutatásával illusztráltuk azt a körülményt, hogy a sztochasztikus optimalizálási problémák konvexitási és(vagy) differenciálhatósági tulajdonságai gyakran nem ismertek, illetve igazolható ezek (valamelyikének) hiánya. A 2.3. részben összefoglalt iteratív sztochasztikus algorit-

musok viszont — a matematikai programozás módszereinek többségéhez hasonló módon — lényegileg csak konvex és(vagy) differenciálható szerkezetű problémák megoldására alkalmazhatók. Miután a nem-konvex (vagyis általában több lokális optimummal rendelkező) feladatok globális optimuma a lokálisan konvergens módszerekkel található megoldások legtöbbjénél lényegesen kedvezőbb lehet, az alábbiakban olyan módszereket vizsgálunk, amelyek alkalmazhatók az említett típusú problémák megoldására, a differenciálhatósági feltételek elhagyásával. Determinisztikus, illetve sztochasztikus globális optimalizálási eljárások áttekintésével foglalkoznak pl. a [15, 38, 94] munkák. Számos determinisztikus módszer alapszik a korlátozás és szétválasztás elvén [22, 24, 65, 79], a lokális optimumhelyek vonzási tartományaiból indított optimalizálási eljárásokon [39, 69, 75], vagy a stacionárius trajektóriák vizsgálatán [8]. A determinisztikus, minimax típusú döntési elven alapuló módszerek bizonyos elvi korlátaira mutatnak rá a [15] Vol. 2, 31—48., [95] dolgozatok. Mivel a megoldandó feladatok sztochasztikus és (esetleg) nem-differenciálható struktúrája a determinisztikus módszerek alkalmazásában további nehézségeket is okozhat, ezért az alábbiakban megfelelő sztochasztikus eljárások kiválasztásával foglalkozunk.

A 3. rész első szakaszában sztochasztikus optimalizálási módszerek egy általános osztályának konvergenciatulajdonságait ismertetjük. A második szakaszban ezeket az eredményeket terjesztjük ki sztochasztikusan kombinált (hibrid) optimalizálási eljárások konvergenciájának igazolására. A leírás során az eredményeket egységes keretben tárgyaljuk, röviden összefoglalva a korábbi dolgozatainkban [48, 52] részletezett vizsgálatokat is.

3.1. *Sztochasztikus optimalizálási eljárások konvergenciatulajdonságai*

Amint azt korábban megjegyeztük, bizonyos típusú sztochasztikus optimalizálási módszerek konvergenciája viszonylag enyhe feltevések mellett biztosítható: a következőkben az említett általános konvergenciatulajdonságokat vizsgáljuk meg. Tegyük fel, hogy a

$$\min_{\mathbf{x}\in C} f(\mathbf{x}) \quad f\colon E^n \to E^1, \quad C \subset E^n \tag{3.1}$$

feladat megoldását közelítő $\{\mathbf{x}_k\}$ pontsorozatot az

$$\mathbf{x}_{k+1} = D(\mathbf{z}_k, \bar{\omega}_k), \quad \mathbf{x}_k \in C, \quad k = 0, 1, 2, \ldots \tag{3.2}$$
$$\mathbf{z}_k = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k\}, \quad \bar{\omega}_k = \{\omega_0, \omega_1, \ldots, \omega_k\}$$

iterációval állítjuk elő. Itt a D sztochasztikus döntési szabály a k-adik lépésben ismert $\mathbf{z}_k$ keresési információtól és a döntést befolyásoló véletlen tényezők sorozatának már realizálódott $\bar{\omega}_k$ szeletétől (k-adik projekciójától) függ. Megjegyezzük, hogy a (3.2) sémával pl. különböző véletlen kereső típusú eljárások [5, 10, 14, 49, 67, 93] is leírhatók: ezekben $\mathbf{x}_{k+1}$ kiválasztása a C megengedett tartományon a $\mathbf{z}_k$ keresési információ figyelembevételével meghatározott (feltételes) valószínűségi mértéknek megfelelően történik.

Vezessük be az $\omega = (\omega_0, \omega_1, \omega_2, \ldots)$, $\omega \in \Omega$ jelöléseket. A továbbiakban feltesszük, hogy a (3.2) sztochasztikus algoritmusnak megfeleltethető egy, az $(\Omega, \mathcal{A})$

mérhető téren értelmezett P valószínűségi mérték. Jelölje $C^* \subset C$, $C^* \neq \emptyset$ valamely (pl. a (3.1)) optimalizálási feladat megoldáspontjainak halmazát. Célunk olyan D döntési szabály (sztochasztikus algoritmus) megválasztása, hogy a (3.2) szerint generált $\{\mathbf{x}_k\}$ sorozatra vonatkozóan fennálljon a

$$P\left(\lim_{k\to\infty} \varrho(\mathbf{x}_k, C^*) = 0\right) = 1, \quad \left(\varrho(\mathbf{x}, C^*) = \inf_{\mathbf{x}^* \in C^*} \|\mathbf{x}-\mathbf{x}^*\|\right)$$

reláció.

Legyen $\{A_k\}$, $A_k \in \mathcal{A}$, $k=0, 1, 2, \ldots$ véletlen eseménysorozat. A sztochasztikus optimalizálási módszerek egy általános osztályának konvergenciája az alábbi egyszerű segédtételek felhasználásával igazolható.

3.1. LEMMA. Ha fennállnak a

$$P\left(A_k \middle| \bigcap_{j=1}^{k} \bar{A}_{k-j}\right) \geqq p_k, \quad 0 \leqq p_k \leqq 1, \quad k = 0, 1, 2, 3, \ldots \tag{3.3}$$

és

$$\prod_{k=0}^{\infty} (1-p_k) = 0 \tag{3.4}$$

vagy

$$\sum_{k=0}^{\infty} p_k = \infty \tag{3.4}'$$

feltételek, akkor igaz a $P\left(\bigcup_{k=0}^{\infty} A_k\right) = 1$ reláció.

Bizonyítás. A (3.3) egyenlőtlenség felhasználásával tetszőleges N természetes szám esetén adódnak a

$$P\left(\bigcap_{k=0}^{N} \bar{A}_k\right) = \prod_{k=0}^{N} P\left(\bar{A}_k \middle| \bigcap_{j=1}^{k} \bar{A}_{k-j}\right) \leqq \prod_{k=0}^{N} (1-p_k) \leqq \exp\left(-\sum_{k=0}^{N} p_k\right)$$

relációk, tehát a (3.4), (3.4)′ feltételek bármelyikének fennállása esetén

$$P\left(\bigcup_{k=0}^{N} A_k\right) \xrightarrow[N\to\infty]{} 1;$$

ezzel a lemmát igazoltuk.

A 3.1. segédtételhez teljesen analóg módon bizonyítható a következő állítás.

3.2. LEMMA. A 3.1. lemmával kapcsolatban bevezetett jelöléseket megtartva tegyük fel, hogy tetszőleges k, n, $k \geqq n$ nem-negatív egész számok esetében fennállnak a

$$P\left(A_k \middle| \bigcap_{j=1}^{k-n} \bar{A}_{k-j}\right) \geqq p_k, \quad 0 \leqq p_k \leqq 1, \quad k \geqq n, \quad k, n = 0, 1, 2, 3, \ldots \tag{3.3}'$$

és (3.4)′ feltételek. Ekkor

$$P(A_\infty) = P\left(\bigcap_{n=0}^{\infty} \bigcup_{k=n}^{\infty} A_k\right) = 1, \tag{3.5}$$

vagyis 1 valószínűséggel az $\{A_k\}$ események közül végtelen sok következik be.

A (3.5) reláció azt mutatja, hogy a 3.2. segédtétel a *Borel—Cantelli-lemma* (l. pl. [63], 323—326.) második állításának egyszerű kiterjesztése, az A_k események függetlenségére vonatkozó feltétel nélkül. Az idézett lemma első állítása szerint a

$$\sum_{k=0}^{\infty} P(A_k) < \infty$$

feltételből következik $P(A_\infty)=0$; most ennek az eredménynek egy, a feltételes valószínűségekre vonatkozó megfelelőjét igazoljuk.

3.3. LEMMA. A korábbi jelöléseket megtartva tegyük fel, hogy tetszőleges $k, n, k \geqq n$ nem-negatív egész számok esetén fennállnak a

$$P\left(A_k \middle| \bigcap_{j=1}^{k-n} \bar{A}_{k-j}\right) \leqq p_k, \quad 0 \leqq p_k < 1, \quad k \geqq n, \quad k, n = 0, 1, 2, 3, \ldots \tag{3.6}$$

és

$$-\sum_{k=0}^{\infty} \ln(1-p_k) < \infty \tag{3.7}$$

feltételek. Ekkor

$$P(A_\infty) = 0, \tag{3.8}$$

vagyis 1 valószínűséggel az A_k események közül csak véges számú következhet be.

Bizonyítás. A (3.6) feltételek következtében bármely n, N $n \leqq N$ nem-negatív egész számpárra igaz

$$P\left(\bigcup_{k=n}^{N} A_k\right) = 1 - P\left(\bigcap_{k=n}^{N} \bar{A}_k\right) \leqq 1 - \prod_{k=n}^{N} (1-p_k),$$

tehát az $N \to \infty$ határátmenet esetén adódik

$$P(A_\infty) \leqq P\left(\bigcup_{k=n}^{\infty} A_k\right) \leqq 1 - \prod_{k=n}^{\infty} (1-p_k), \quad n = 0, 1, 2, 3, \ldots. \tag{3.9}$$

Mivel (3.7) értelmében $\sum_{k=n}^{\infty} \ln(1-p_k) \xrightarrow[n\to\infty]{} 0$, vagyis $\ln \prod_{k=n}^{\infty} (1-p_k) \xrightarrow[n\to\infty]{} 0$, adódik, hogy $\prod_{k=n}^{\infty} (1-p_k) \xrightarrow[n\to\infty]{} 1$. Ezért (3.9)-ből következik a bizonyítandó (3.8) reláció.

Megjegyzések

1. Egyszerűen belátható, hogy (3.7) ekvivalens a $\sum_{k=0}^{\infty} p_k < \infty$ feltétellel.

2. A 3.2. és 3.3. lemmák alapján adaptív sztochasztikus algoritmusok konvergenciájának elégséges, illetve szükséges feltételei adódnak. A 3.2. lemma különböző módszerek (pl. [5], [14], [31], [32], [48], [67], [74], [78], [93]) konvergenciájának bizonyítására felhasználható. Az említett algoritmusok közül néhány különböző ismert determinisztikus matematikai programozási eljárások és valamilyen egyszerű véletlen keresési módszer kombinációján alapszik. A 3.3. lemma alkalmazását illuszt-

rálja az a korábbi dolgozatunkban ([48], 210—212.) leírt függvényminimalizálási példa, amelyben a (3.6)—(3.7) típusú relációk fennállása esetén az ott leírt (egyéb feltételek mellett konvergens) véletlen kereső eljárás nem konvergál a függvény minimumhelyéhez.

3. Hangsúlyozzuk, hogy a 3.2. lemma a (3.1) feladatra vonatkozó további feltevések (pl. *f Lipschitz-folytonos, C* korlátos) híján nem teszi lehetővé az optimális megoldás becslését a $\mathbf{z}_k$, $k=0, 1, 2, 3, \ldots$ kereső információ ismeretében: ez a sztochasztikus algoritmusoknak általános jellemzője. Megjegyezzük még, hogy pl. a

$$P\left(A_k \middle| \bigcap_{j=1}^{k} \bar{A}_{k-j}\right) = P(A_k) = p, \quad k = 0, 1, 2, 3, \ldots \quad (0 < p < 1) \tag{3.10}$$

speciális esetben (egymástól függetlenül kiválasztott, azonos eloszlású, véletlen mintapontok esete) a

$$P\left(\bigcup_{k=0}^{N} A_k\right) \geqq 1-\varepsilon, \quad (0 < \varepsilon < 1) \quad \text{reláció fennállásához} \quad N \geqq \frac{\ln \varepsilon}{\ln (1-p)} - 1$$

szükséges. Ebből az is látható, hogy N a gyakorlatilag érdekes $p \to 0$, $\varepsilon \to 0$ esetekben igen gyorsan növekszik: ez a tény az egyszerű véletlen kereső eljárásoknak egyéb optimalizálási módszerekkel való kombinációjának szükségességére mutat rá. Mielőtt ezt a lehetőséget megvizsgálnánk, a sztochasztikus optimalizálási eljárások néhány alaptípusának konvergenciáját igazoljuk a 3.2. lemma felhasználásával.

Tekintsük először a 3.1.—3.3. lemmákkal kapcsolatban bevezetett általános feladatot: előállítandó olyan $\{\mathbf{x}_k\} \subset C$ (véletlen) pontsorozat, amelynek torlódási pontjai a $C \subset E^n$ megengedett tartomány $C^* \neq \emptyset$ részhalmazába esnek (1 valószínűséggel). Legyen adott egy $f: C \to E^1$ indikátorfüggvény. Az $\{\mathbf{x}_k\}$ pontsorozat előállítására az alábbi típusú sztochasztikus eljárást alkalmazzuk:

0. Legyen $\mathbf{x}_0 \in C$ tetszőleges, $k := 0$.

1. $k=0, 1, 2, \ldots$ esetén egy újabb pontot az

$$\mathbf{y} = D(\mathbf{z}_k, \bar{\omega}_k), \quad \mathbf{y} \in C \tag{3.11}$$

$$\mathbf{z}_k = \{\mathbf{x}_j, f(\mathbf{x}_j), \quad j = 0, 1, \ldots, k\} \quad \bar{\omega}_k = \{\omega_0, \omega_1, \ldots, \omega_k\}$$

előírás szerint állítunk elő (vö. (3.2)).

2. Az $\{\mathbf{x}_k\}$ pontsorozat következő elemét az

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{y}, & \text{ha } f(\mathbf{y}) < f(\mathbf{x}_k) \\ \mathbf{x}_k, & \text{ha } f(\mathbf{y}) \geqq f(\mathbf{x}_k) \end{cases} \tag{3.12}$$

szabállyal definiáljuk, ezután ($k := k+1$ értékadással) az 1. lépésre térünk vissza.

Tetszőleges $\delta > 0$ esetén legyen

$$G(C^*, \delta) = \{\mathbf{y} \in C: \varrho(\mathbf{y}, C^*) < \delta\},$$

$$\hat{C} = \hat{C}(\delta) = C \setminus G(C^*, \delta) = \{\mathbf{y} \in C: \varrho(\mathbf{y}, C^*) \geqq \delta\}.$$

Jelölje tetszőleges $\hat{\varepsilon} > 0$ esetén A_k az

$$A_k(\hat{\varepsilon}): \{f(\mathbf{x}_{k+1}) \leqq f(\mathbf{x}_k) - \hat{\varepsilon}\}$$

véletlen eseményt, legyen továbbá

$$F_k = F_k(\hat{\delta})\colon \{\mathbf{x}_k \in \hat{C}\}.$$

3.1. TÉTEL. Tegyük fel, hogy fennállnak a következő feltételek:

1. f alulról korlátos, mérhető függvény a C halmazon, C^* tartalmazza f minimumhelyeit.

2. Tetszőleges $\hat{\delta}>0$ esetén létezik olyan $\hat{\varepsilon}=\hat{\varepsilon}(\hat{\delta})>0$ szám és $\{\hat{p}_k\}$ $\hat{p}_k=\hat{p}_k(\hat{\delta})$, $0\leqq\hat{p}_k\leqq 1$ számsorozat, hogy $k, n=0, 1, 2, \ldots$ $k\geqq n$ esetén fennáll

$$P\left(A_k \cup \bar{F}_k \middle| \bigcap_{j=1}^{k-n} (\bar{A}_{k-j} \cap F_{k-j})\right) \geqq \hat{p}_k. \tag{3.13}$$

3. Bármely $\hat{\delta}>0$ esetén a $\{\hat{p}_k\}$ számsorozat kielégíti a

$$\sum_{k=0}^{\infty} \hat{p}_k = \infty \tag{3.14}$$

feltételt.

Ekkor

$$P(\liminf_{k\to\infty} \varrho(\mathbf{x}_k, C^*) = 0) = 1.$$

Bizonyítás. Elég megmutatni, hogy tetszőleges $\delta>0$ esetén fennáll

$$P(F(\delta))=0, \quad \text{ahol} \quad F(\delta)\colon \{\omega\colon \liminf_{k\to\infty} \varrho(\mathbf{x}_k, C^*) \geqq \delta\}$$

Tegyük fel, hogy ez nem igaz, tehát valamilyen $\delta>0$ mellett

$$P(\omega\colon \liminf_{k\to\infty} \varrho(\mathbf{x}_k, C^*) \geqq \delta) = p, \quad p = p(\delta) > 0.$$

Ez azt jelenti, hogy a szóban forgó ω-kra teljesül $\varrho(\mathbf{x}_k, C^*)\geqq\frac{\delta}{2}$, ha $k\geqq k_0=k_0(\delta, \omega)$.

Legyen $\hat{\delta}=\frac{\delta}{2}$, akkor $F(\delta)\subseteq \bigcup_{n=0}^{\infty} \bigcap_{k=n}^{\infty} F_k(\hat{\delta})$. A tétel 2.—3. feltételei szerint az adott $\hat{\delta}>0$ számhoz létezik $\hat{\varepsilon}>0$ és $\{\hat{p}_k\}$ $0\leqq\hat{p}_k\leqq 1$ úgy, hogy $0\leqq n\leqq k$ esetén fennáll (3.13) és (3.14). A 3.2. lemmát alkalmazva adódik

$$P(\bar{A}_\infty \cap F(\delta))=0, \quad \text{tehát} \quad P(A_\infty | F(\delta))=1.$$

Így azt kapjuk, hogy ha igaz $\liminf_{k\to\infty} \varrho(\mathbf{x}_k, C^*)\geqq\delta$, akkor az A_k események közül végtelen sok bekövetkezik, tehát — (3.12) és $\hat{\varepsilon}$ pozitív volta miatt — érvényes a

$$P(\omega\colon \lim_{k\to\infty} f(\mathbf{x}_k) = -\infty \,|\, \liminf_{k\to\infty} \varrho(\mathbf{x}_k, C^*) \geqq \delta) = 1$$

egyenlőség. Így

$$P(\omega\colon \lim_{k\to\infty} f(\mathbf{x}_k) = -\infty) \geqq p(\delta) > 0;$$

a kapott reláció viszont ellentmond a tétel 1. feltételének, amely szerint f alulról korlátos C-n. Ezért $p(\delta)=0$ bármely $\delta>0$ esetén; ezzel a tétel állítását igazoltuk.

3.1. KÖVETKEZMÉNY. Ha

$$C^* = \{\mathbf{x}^* \in C\colon f(\mathbf{x}^*) = \min_{\mathbf{x}\in C} f(\mathbf{x})\}, \quad C^* \neq \emptyset,$$

f egyenletesen folytonos C-n, és bármely pozitív $\delta>0$ esetén létezik olyan $\varepsilon=\varepsilon(\delta)>0$ szám, hogy az

$$\mathbf{x}\in C\setminus G(C^*,\delta)=\{\mathbf{x}\in C:\ \varrho(x,C^*)\geqq\delta\}$$

relációból következik

$$f(\mathbf{x})\geqq f(\mathbf{x}^*)+\varepsilon,$$

akkor a 3.1. tétel feltételeinek fennállása esetén

$$P\big(\lim_{k\to\infty}\varrho(\mathbf{x}_k,C^*)=0\big)=1,$$

ezért az $\{\mathbf{x}_k\}$ sorozat minden határpontja 1 valószínűséggel a C^* halmazhoz tartozik.

Bizonyítás. Megmutatjuk, hogy

$$P(\omega:\ \limsup_{k\to\infty}\varrho(\mathbf{x}_k,C^*)=0)=1.$$

Legyen $f^*=\min_{\mathbf{x}\in C}f(\mathbf{x})$. Az f egyenletes folytonossága alapján felírhatók a következő relációk:

(3.15)
$$1=P(\omega:\ \liminf_{k\to\infty}\varrho(\mathbf{x}_k,C^*)=0)=$$

$$=P(\omega:\ \lim_{i\to\infty}\varrho(\mathbf{x}_{k_i},C^*)=0\quad\text{alkalmas }\{\mathbf{x}_{k_i}\}\subset\{\mathbf{x}_k\}\text{ esetén})=$$

$$=P(\omega:\ \lim_{i\to\infty}f(\mathbf{x}_{k_i})=f^*)=P(\omega:\ \lim_{k\to\infty}f(\mathbf{x}_k)=f^*).$$

Tegyük fel most, hogy valamely $\nu>0$ esetén

$$P(\omega:\ \limsup_{k\to\infty}\varrho(\mathbf{x}_k,C^*)\geqq\nu)=p,\quad p=p(\nu)\geqq 0,$$

akkor ezekre az ω-kra létezik $\{\mathbf{x}_{k_j}\}\subset\{\mathbf{x}_k\}$ részsorozat úgy, hogy $\varrho(\mathbf{x}_{k_j},C^*)\geqq\delta=\frac{\nu}{2}$, $\{k_j(\omega)\}\subseteqq\{k\}$. Ezért megfelelő $\varepsilon=\varepsilon(\delta)>0$ esetén $f(\mathbf{x}_{k_j})\geqq f^*+\varepsilon$: ez viszont ellentmond az 1 valószínűséggel teljesülő

$$\lim_{j\to\infty}f(\mathbf{x}_{k_j})=\lim_{k\to\infty}f(\mathbf{x}_k)=f^*\quad\text{(l. (3.15))}$$

relációknak; tehát $p(\nu)=0$ minden $\nu>0$ mellett. Ezzel a következményt igazoltuk.

Megjegyzés. A 3.1. következménynek az $\varepsilon=\varepsilon(\delta)>0$ számok létezésére vonatkozó feltétele a feladat bizonyos értelemben vett regularitását írja elő. Ez a feltétel nem teljesül pl. a

$$C=[0,\infty),\quad f(x)=\begin{cases}x^2, & \text{ha}\quad 0\leqq x\leqq 1\\ \dfrac{1}{x}, & \text{ha}\quad x>1\end{cases}$$

esetben. Nem nehéz ekkor olyan (3.11)—(3.12) típusú algoritmust megadni, hogy az általa generált $\{\mathbf{x}_k\}$ pontsorozat kielégítse a

$$P(\liminf_{k\to\infty}\varrho(\mathbf{x}_k,C^*)=0)=P(\limsup_{k\to\infty}\varrho(\mathbf{x}_k,C^*)=\infty)=1$$

relációkat.

A 3.1. tétel alkalmazásaként adaptív sztochasztikus optimalizálási eljárások globális konvergenciájára vonatkozó állítást igazolunk. Legyen $C \subset E^n$ valamely nem-üres, korlátos, nyílt halmaz lezártja, $f: E^n \to E^1$ a C halmazon konstanstól különböző folytonos célfüggvény. Legyen

$$C^* = \{\mathbf{x}^* \in C: f^* = f(\mathbf{x}^*) = \min_{\mathbf{x} \in C} f(\mathbf{x})\}$$

a (3.1) feladat megoldásainak halmaza (a feltételek értelmében C nem-üres, korlátos. zárt halmaz). Tegyük fel, hogy a

$$\bar{G}(C^*, \delta) = \{\mathbf{x} \in C: \varrho(\mathbf{x}, C^*) \leqq \delta\}$$

halmazok *Lebesgue-mértéke* pozitív, ha $\delta > 0$.

A (3.1) feladat megoldására a következő típusú sztochasztikus algoritmust alkalmazzuk:

0. Legyen $\mathbf{x}_0 \in C$ tetszőleges pont, $k = 0$.

1. A $(k+1)$-edik lépésben előállítjuk az

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma_k \mathbf{s}_k, \quad \mathbf{s}_k \sim \mathbf{P}_k \tag{3.16}$$

pontot: itt $\mathbf{s}_k = \mathbf{s}_k(\mathbf{z}_k, \bar{\omega}_k) \in E^n$ az iterációs lépés iránya, $\gamma_k = \gamma_k(\mathbf{z}_k, \bar{\omega}_k) \geqq 0$ pedig a lépés hossza. Az $\mathbf{s}_k$ irányt egy valószínűségi vektorváltozó realizációjaként állítjuk elő és feltesszük, hogy az $\mathbf{s}_k$ eloszlásfüggvényének megfelelő $\mathbf{P}_k$ valószínűségi mérték a C halmazon ekvivalens (vagyis kölcsönösen abszolút folytonos) a $\mu^{(n)}$ n-dimenziós *Lebesgue-mértékkel*, továbbá $\liminf_{k \to \infty} P_k(C_1) > 0$, ha $C_1 \subset C$ és $\mu^{(n)}(C_1) > 0$.

2. A γ_k lépéshossz meghatározása a kiválasztott irányban végrehajtott globális kereséssel történik:

$$f(\mathbf{x}_{k+1}) = \min_{\substack{\gamma \geqq 0 \\ \mathbf{x}_k + \gamma \mathbf{s}_k \in C}} f(\mathbf{x}_k + \gamma \mathbf{s}_k). \tag{3.17}$$

3.2. TÉTEL.

$$P\left(\lim_{k \to \infty} \varrho(\mathbf{x}_k, C^*) = 0\right) = 1,$$

tehát a (3.16)—(3.17) sztochasztikus algoritmus által generált $\{\mathbf{x}_k\}$ sorozat minden határpontja 1 valószínűséggel a C^* halmazhoz tartozik.

Bizonyítás. Az állítást a 3.1. tétel és következménye felhasználásával bizonyítjuk. Az f célfüggvény folytonos a kompakt C halmazon, ezért korlátos is. A továbbiakban a 3.1. tétel 2. és 3. feltételének teljesülését ellenőrizzük. Legyen $\hat{\delta}_1 = \hat{\delta} > 0$, a korábbi módon definiáljuk a

$$G(C^*, \hat{\delta}) = \{\mathbf{y} \in C: \varrho(\mathbf{y}, C^*) < \hat{\delta}\}, \quad \hat{C} = C \setminus G(C^*, \hat{\delta})$$

halmazokat. Esetünkben $\hat{C}$ — korlátos, zárt halmaz, legyen $\hat{f} = \min_{\mathbf{x} \in \hat{C}} f(\mathbf{x})$, $2\hat{\varepsilon} = \hat{f} - f^* > 0$. Az f függvény folytonossága miatt létezik olyan $\hat{\delta}_2 = \hat{\delta}_2(\hat{\varepsilon}) > 0$, $\hat{\delta}_2 < \hat{\delta}_1$ szám, hogy

$$\mathbf{y} \in \bar{G} = \bar{G}(C^*, \hat{\delta}_2) = \{\mathbf{y} \in C: \varrho(\mathbf{y}, C^*) \leqq \hat{\delta}_2\}$$

esetén fennáll az

$$f(\mathbf{y}) \leqq f^* + \hat{\varepsilon}$$

egyenlőtlenség. Ezért tetszőleges $\mathbf{x}\in\hat{C}$ és $\mathbf{y}\in\bar{G}$ esetén

$$f(\mathbf{y}) \leqq f(\mathbf{x})-\hat{\varepsilon}.$$

$k=0, 1, 2, \ldots$ esetén definiáljuk az A_k, F_k, B_k véletlen eseményeket a következő módon:

$$A_k\colon \{f(\mathbf{x}_{k+1}) \leqq f(\mathbf{x}_k)-\hat{\varepsilon}\}, \quad F_k\colon \{\mathbf{x}_k\in\hat{C}\},$$

B_k: {létezik olyan $\gamma\geqq 0$, amelyre $\mathbf{x}_k+\gamma\mathbf{s}_k$ a $\bar{G}$ halmaz eleme}.

A (3.17) iránymenti optimalizálási stratégia értelmében $B_k\cap F_k\subseteqq A_k\cap F_k$, így $0\leqq n\leqq k$ esetén fennáll

$$P\left(A_k\cup\bar{F}_k\middle|\bigcap_{j=1}^{k-n}(\bar{A}_{k-j}\cap F_{k-j})\right) \geqq P\left(B_k\cup\bar{F}_k\middle|\bigcap_{j=1}^{k-n}(\bar{A}_{k-j}\cap F_{k-j})\right).$$

Itt az $\mathbf{s}_k$ irányok megválasztására vonatkozó feltétel és a $\bar{G}$ halmaz pozitív *Lebesgue-mértéke* miatt létezik olyan $\hat{p}=\hat{p}(\hat{\delta})>0$ szám, amellyel teljesülnek a

$$P\left(B_k\cup\bar{F}_k\middle|\bigcap_{j=1}^{k-n}(\bar{A}_{k-j}\cap F_{k-j})\right) \geqq \hat{p}_k \equiv \hat{p}, \quad k, n = 0, 1, 2, \ldots, \quad k \geqq n$$

relációk, tehát a 3.1. tétel 2. és 3. feltételei is fennállnak. A bizonyításból az is látható, hogy bármely $\delta>0$ esetén létezik olyan $\varepsilon=\varepsilon(\delta)>0$, hogy $\mathbf{x}\in C\setminus G(C^*, \delta)$ maga után vonja az $f(\mathbf{x})\geqq f^*+\varepsilon$ egyenlőtlenséget. Ezért a 3.1. tétel és következménye alkalmazhatók, tehát adódik a bizonyítandó

$$P\left(\lim_{k\to\infty}\varrho(\mathbf{x}_k, C^*)=0\right)=1$$

reláció.

Megjegyzések

1. A 3.2. tételben leírt algoritmus-séma realizálható pl. úgy, hogy az $\mathbf{s}_k$ kereső irányokat az E^n tér egységgömbjének felületén egyenletes eloszlású véletlen vektorok független kiválasztásaival határozzuk meg, az iránymenti minimalizálást pedig valamilyen globális optimalizálási módszerrel végezzük. Számos olyan módszer ismeretes (l. pl. [65], [79], [94]), amely pl. véges számú diszjunkt szakaszon értelmezett *Lipschitz-folytonos* függvény esetén alkalmas a (3.17) feltételt kielégítő iránymenti optimum közelítő meghatározására. Ebből az is látható, hogy a megengedett tartomány konvexitása nem csak a bizonyításhoz, hanem a numerikus realizációhoz sem feltétlenül szükséges (bár az utóbbit természetesen egyszerűbbé teszi).

2. A 3.2. tétel könnyen kiterjeszthető arra az esetre is, amikor a C megengedett tartománynak és az f célfüggvénynek az iteráció során csak megfelelő módon fokozatosan pontosított közelítései ismeretesek. Ezt a lehetőséget a 3.2. szakaszban általánosabb keretek között fogjuk tárgyalni.

A továbbiakban véletlen kereső eljárások lokális konvergenciatulajdonságait vizsgáljuk. Ennek kapcsán egy konvergenciatétel kimondására szorítkozunk; ez valamivel általánosabb a korábbi dolgozatunkban igazolt állításnál ([48], 2.2. tétel), de lényegileg azonos eszközökkel bizonyítható.

Tekintsük a (3.1) feladatot a következő feltevések mellett:

1. C egy összefüggő, nyílt halmaz lezártja.

2. $f(\mathbf{x}) > -\infty$ a C halmazon adott folytonos függvény, amelynek a C-be eső lokális minimumhelyei a nem-üres C^* halmazhoz tartoznak; C^* véges elemszámú.

3. Ha $\hat{\mathbf{x}} \in C \setminus C^*$, akkor létezik olyan $\hat{H} = \hat{H}(\hat{\mathbf{x}}) \subset E^n$ zárt konvex halmaz, amelynek $\hat{\mathbf{x}}$ érintési pontja, $\mu^{(n)}(\hat{H} \cap C) > 0$, és az $\mathbf{y} \in \hat{H} \setminus \{\hat{\mathbf{x}}\}$ pontokban fennáll $f(\mathbf{y}) < f(\hat{\mathbf{x}})$.

4. A (3.1) feladat lokális minimumpontjainak meghatározására az

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma_k \mathbf{s}_k, \quad k = 0, 1, 2, 3, \ldots$$

$$f(\mathbf{x}_{k+1}) \leqq \min_{\substack{0 \leqq \gamma \leqq h_k \\ \mathbf{x}_k + \gamma \mathbf{s}_k \in C}} f(\mathbf{x}_k + \gamma \mathbf{s}_k) + \varepsilon_k \tag{3.18}$$

feltételek szerinti iterációt alkalmazzuk. Itt $\mathbf{x}_0 \in C$ tetszőleges kezdőpont, $\mathbf{s}_k \in E^n$ a (3.16) relációval kapcsolatban ismertetett tulajdonságú P_k mérték szerint kerül kiválasztásra, ezután normáljuk: $\|s_k\| = 1$, $k = 0, 1, 2, 3, \ldots$.

A (3.18) iránymenti lokális optimalizálások hatósugara alulról korlátos: $h_k \geqq h > 0$, $k = 0, 1, 2, 3, \ldots$, az ε_k véletlen hibák sorozatára vonatkozóan pedig 1 valószínűséggel teljesül $\sum_{k=0}^{\infty} \varepsilon_k \leqq E$ ($0 < E < \infty$ állandó).

Érvényes a következő állítás.

3.3. TÉTEL. Az 1.—4. feltételek fennállása esetén a generált $\{\mathbf{x}_k\}$ pontsorozat minden határpontja 1 valószínűséggel a C^* halmazhoz tartozik.

Az idézett [48] dolgozatban a 3.3. tétel 3. és 4. feltételeinek megfelelőit (szükségesség, számítógépi realizálhatóság) részletesen vizsgáltuk. Megjegyezzük még, hogy a 3.1 tétel és következménye analógiájára is bizonyítható lokális konvergenciaeredmény.

3.2. *Sztochasztikusan kombinált optimalizálási módszerek konvergenciatulajdonságai*

Amint az előző szakaszban leírt 3.1.—3.3. tételekből kitűnik, a véletlen kereső típusú sztochasztikus optimalizálási eljárások meglehetősen általános feltételek mellett rendelkeznek globális, illetve lokális konvergenciatulajdonságokkal; ugyanakkor utaltunk arra is, hogy az ilyen módszerek numerikus hatékonysága (különösen a minimumhely(ek) környezetében) gyakran elég kicsiny. Ezért optimalizálási feladatok megoldása során célszerű lehet a kezdetben viszonylag hatékony, egyszerűbb szerkezetű véletlen kereső módszereket a megoldás(ok) közelében valamilyen lokálisan hatékony eljárással felcserélni. (Ismeretes pl., hogy ha a (3.1) feladat célfüggvénye a minimumhely(ek) környezetében kvadratikusan jól közelíthető, akkor pl. a konjugált gradiens típusú eljárások lokálisan igen hatékonyak lehetnek [20], [21].) Különböző optimalizálási eljárásoknak a fenti típusú heurisztikus kombinációit korábban is gyakran alkalmazták pl. műszaki, gazdasági tervezésben felmerülő nemkonvex optimalizálási feladatok megoldására (ezzel kapcsolatban l. pl. a [14], [15], [16], [39], [50], [69], [75], [94] munkákat). Érdemes hangsúlyozni, hogy az említett eljárások konvergenciáját a lokálisan hatékony módszerek még differenciálható,

determinisztikus, de nem-konvex optimalizálási feladatok esetében is csak további speciális feltételek mellett biztosítják, ezért a konvergenciát a módszerekben kellő súllyal bíró sztochasztikus részalgoritmussal lehet elérni. A sztochasztikus módszereknek a 3.1.-ben leírt tulajdonságai viszont a konvergenciát nem-konvex, nem-differenciálható (folytonos) és sztochasztikus optimalizálási feladatok esetében is biztosítják.

A [14] és [48] dolgozatok eredményeit felhasználva és általánosítva [52] dolgozatunkban bevezettük sztochasztikusan kombinált (hibrid) optimalizálási eljárások egy általános osztályát és igazoltuk a hibrid módszerek lokális és globális konvergenciatulajdonságait. A következőkben az ott leírt eredményeket általánosabb alakban összegezzük. A kimondott lokális és globális konvergenciatételek bizonyítása lényegileg megegyezik az [52] dolgozatban leírt 2.2. és 3.2. tételekével, ezért a tételek igazolását mellőzzük.

A lokális és globális konvergenciaeredmények bizonyítása a 3.1.—3.2. lemmák alábbi általánosításain alapszik. Ismét a (3.1) feladat megoldásainak $C^* \subset C, C^* \neq \emptyset$ halmazához 1 valószínűséggel konvergáló véletlen $\{\mathbf{x}_k\}$ sorozat előállítását tűzzük ki célul. A 3.1. lemmával kapcsolatban bevezetett jelöléseket használjuk: legyen P a vizsgált sztochasztikus algoritmusnak megfelelő valószínűségi mérték az $(\Omega, \mathcal{A})$ mérhető téren, $\{A_k\}$ pedig egy eseménysorozat. A (3.1) feladat megoldására sztochasztikusan kombinált módszert alkalmazunk, amely a következőképpen írható le. Legyen $\mathbf{x}_0 \in C$ az algoritmus tetszőleges kiindulópontja, az iterációs lépések során az újabb közelítő megoldást előállító döntési szabály:

$$\mathbf{x}_{k+1} = \mathbf{H}_k(\mathbf{z}_k) \quad \mathbf{z}_k = \{\mathbf{x}_j, f(\mathbf{x}_j),\ j = 0, 1, \ldots, k\} \tag{3.19}$$

$$\mathbf{H}_k\colon\ \alpha_k \mathbf{G}_k(\mathbf{z}_k) + (1-\alpha_k)\,\mathbf{L}_k(\mathbf{z}_k) \quad 0 \leqq \alpha_k \leqq 1, \quad k = 0, 1, 2, 3, \ldots.$$

Ez azt jelenti, hogy a k-adik lépésben — az előző lépések eredményétől függetlenül — α_k valószínűséggel a $\mathbf{G}_k$, $(1-\alpha_k)$ valószínűséggel pedig az $\mathbf{L}_k$ szubalgoritmus kerül végrehajtásra; $\mathbf{G}_k$ és $\mathbf{L}_k$ általában a korábbi iterációs lépések során nyert információtól függ, emellett az $\bar{\omega}_k = (\omega_0, \omega_1, \ldots, \omega_k)$ véletlen tényezőktől is függhet.

Érvényes a következő segédtétel.

3.4. LEMMA. Ha a $\mathbf{G}_k$ szubalgoritmus alkalmazása biztosítja a

$$P\left(A_k \middle| \bigcap_{j=1}^{k} \bar{A}_{k-j};\ \mathbf{G}_k\right) \geqq p_k, \quad 0 \leqq p_k \leqq 1, \quad k = 0, 1, 2, 3, \ldots \tag{3.20}$$

egyenlőtlenségek fennállását, ahol $A_k = A_k(\bar{\omega}_k)$ és

$$\prod_{k=0}^{\infty} (1-\alpha_k p_k) = 0 \tag{3.21}$$

vagy

$$\sum_{k=0}^{\infty} \alpha_k p_k = \infty, \tag{3.21}'$$

akkor igaz a

$$P\left(\bigcup_{k=0}^{\infty} A_k\right) = 1 \tag{3.22}$$

reláció.

Bizonyítás. A teljes valószínűség tétele és a lemma (3.20) feltétele alapján adódik

$$P\left(A_k \middle| \bigcap_{j=1}^{k} \bar{A}_{k-j}\right) = \alpha_k P\left(A_k \middle| \bigcap_{j=1}^{k} \bar{A}_{k-j};\ \mathbf{G}_k\right) + (1-\alpha_k) P\left(A_k \middle| \bigcap_{j=1}^{k} \bar{A}_{k-j};\ \mathbf{L}_k\right) \geqq \alpha_k p_k,$$

$$k = 0, 1, 2, 3, \ldots$$

ezért bármely N természetes szám esetén igaz

$$P\left(\bigcap_{k=0}^{N} \bar{A}_k\right) = \prod_{k=0}^{N} P\left(\bar{A}_k \middle| \bigcap_{j=1}^{k} \bar{A}_{k-j}\right) \leqq \prod_{k=0}^{N} (1-\alpha_k p_k) \leqq \exp\left(-\sum_{k=0}^{N} \alpha_k p_k\right).$$

Így a (3.21), ill. (3.21)′ feltételek alapján adódik a bizonyítandó (3.22) reláció.

A 3.4. segédtételhez teljesen analóg módon igazolható a következő állítás.

3.5. LEMMA. A 3.4. lemmával kapcsolatban bevezetett jelöléseket megtartva tegyük fel, hogy tetszőleges $k, n, k \geqq n$ nem-negatív egész számok esetén fennállnak a

$$(3.23)\quad P\left(A_k \middle| \bigcap_{j=1}^{k-n} \bar{A}_{k-j};\ \mathbf{G}_k\right) \geqq p_k, \quad 0 \leqq p_k \leqq 1, \quad k \geqq n, k, n = 0, 1, 2, 3, \ldots$$

és

$$(3.21)'\qquad \sum_{k=0}^{\infty} \alpha_k p_k = \infty$$

feltételek. Ekkor

$$P(A_\infty) = 1.$$

Megemlítjük, hogy a 3.3. segédtétel alapján a jelen esetben is megfogalmazható az optimalizálási eljárás konvergenciájára vonatkozó szükséges feltétel.

A 3.4., ill. 3.5. lemmák felhasználásával a hibrid eljárások következő lokális és globális konvergenciatulajdonságai igazolhatók (vö. [52]). Tegyük fel, hogy a (3.1) feladat megoldására irányuló iteráció során mind a megengedett megoldások C halmaza, mind pedig az f célfüggvény értékei csak bizonyos pontatlansággal ismertek, azaz legyen

$$(3.24)\qquad f(\mathbf{x}_k) \approx f(\mathbf{x}_k) + \xi(\mathbf{x}_k)$$

$$C \quad \approx C \quad + \eta(\mathbf{x}_k) \quad k = 0, 1, 2, 3, \ldots.$$

A lokális konvergenciatulajdonságokat a következő feltételek mellett vizsgáljuk:

1. a) $f(\mathbf{x})$ folytonos függvény, amely értelmezve van egy, a C-t tartalmazó C_0 nyílt halmazon.

b) $\lim_{\|\mathbf{x}\| \to \infty} f(\mathbf{x}) = \infty$.

c) Ha $\hat{\mathbf{x}} \in C$ az f függvénynek nem lokális minimumhelye, akkor létezik olyan $\hat{H} = \hat{H}(\hat{\mathbf{x}}) \subset E^n$ zárt konvex halmaz, amelynek $\hat{\mathbf{x}}$ érintési pontja, $\mu^{(n)}(\hat{H} \cap C) > 0$ és az $\mathbf{y} \in \hat{H} \setminus \{\hat{\mathbf{x}}\}$ pontokban fennáll $f(\mathbf{y}) < f(\hat{\mathbf{x}})$.

2. a) $\xi(\mathbf{x}_k) = \xi(\mathbf{x}_k, \bar{\omega}_k)$ valószínűségi változó, vagyis egy megadott (Ω, A, P) mértéktéren értelmezett, az $\mathbf{x}_k$ paramétertől függő P-mérhető függvény.

b) Létezik olyan $0 < S < \infty$ állandó, amellyel teljesül $\left|\sum_{k=0}^{\infty} \xi(\mathbf{x}_k, \bar{\omega}_k)\right| \leqq S$. Hang-

súlyozzuk, hogy ezt és a további hasonló feltételek fennállását a generált $\mathbf{x}_k$ sorozattól függetlenül (ω-ban egyenletesen) és 1 valószínűséggel követeljük meg.

3. $C \subset E^n$ egy nem-üres, összefüggő, nyílt halmaz lezártja.

4. a) $\eta(\mathbf{x}_k) = \eta(\mathbf{x}_k, \bar{\omega}_k)$ egy P-mérhető véletlen pont—halmaz leképezés, vagyis $\eta: E^n \times \Omega \to 2^{E^n}$ (bármely $\mathbf{x} \in E^n$ és $\omega \in \Omega$ esetén $\eta(\mathbf{x}, \omega)$ E^n egy részhalmaza).

b) Az $\eta(\mathbf{x}_k, \bar{\omega}_k)$ halmaz normáját az $\|\eta(\mathbf{x}_k, \bar{\omega}_k)\| = \operatorname{vrai\,sup}_{\mathbf{z} \in \eta(\mathbf{x}_k, \bar{\omega}_k)} \|\mathbf{z}\|$ egyenlőséggel definiálva teljesül

$$\lim_{k \to \infty} \|\eta(\mathbf{x}_k, \bar{\omega}_k)\| = 0.$$

Megjegyezzük, hogy az 1. és 3. feltételek a „beágyazott" determinisztikus feladat típusát írják le, míg 2. és 4. a célfüggvénynek és a megengedett tartománynak az iteráció során fokozatosan pontosított meghatározása egy modelljét adják meg. Az utolsó feltétel az alkalmazható hibrid eljárás típusát definiálja:

5. Az $\{\mathbf{x}_k\}$ véletlen pontsorozatot az $\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma_k \mathbf{s}_k$ iterációval állítjuk elő, ahol az $\mathbf{s}_k \in E^n$, $\|s_k\| = 1$ keresési irányt és a $\gamma_k \geqq 0$ lépéshosszt az alábbi előírások szerint állítjuk elő:

$$\mathbf{s}_k\colon\ \beta_k \mathbf{R}_k + (1-\beta_k)\,\mathbf{L}_k, \quad \beta \leqq \beta_k \leqq 1, \qquad (\beta > 0 \text{ állandó})$$

$$\gamma_k\colon\ f(\mathbf{x}_{k+1}) + \xi(\mathbf{x}_{k+1}) \leqq \min_{\substack{0 \leqq \gamma \leqq h_k \\ \mathbf{x}_k + \gamma \mathbf{s}_k \in C + \eta(\mathbf{x}_k) \\ h_k \geqq h > 0}} f(\mathbf{x}_k + \gamma \mathbf{s}_k) + \varepsilon(\mathbf{x}_k) \qquad (h > 0 \text{ állandó})$$

$$\varepsilon(\mathbf{x}_k) = \varepsilon(\mathbf{x}_k, \bar{\omega}_k) \geqq 0, \quad \sum_{k=0}^{\infty} \varepsilon(\mathbf{x}_k) \leqq E \quad (0 < E < \infty \text{ állandó}).$$

A fenti formulák szerint $\mathbf{s}_k$ előállítására az $\mathbf{R}_k$ és $\mathbf{L}_k$ szubalgoritmusok sztochasztikus kombinációját alkalmazzuk: itt $\mathbf{R}_k$ egy, a 3.3 tételnél (s_k generálásával kapcsolatban) leírtaknak megfelelő eloszlásfüggvényű n-dimenziós valószínűségi változót realizáló, $\mathbf{L}_k$ pedig elvileg tetszőleges eljárás. A γ_k lépéshossz kiválasztása közelítő lokális iránymenti minimalizálással történik: a közelítések $\varepsilon_k(\mathbf{x}_k, \bar{\omega}_k)$ maximális véletlen hibái fokozatosan csökkennek.

Megjegyezzük, hogy az 1.—5. feltételekben szereplő $\xi(\mathbf{x}_k, \bar{\omega}_k)$, $\eta(\mathbf{x}_k, \bar{\omega}_k)$, $\varepsilon(\mathbf{x}_k, \bar{\omega}_k)$ valószínűségi változókra, ill. véletlen leképezésekre vonatkozó (az iteratív sztochasztikus algoritmusok konvergenciabizonyításainál szokásos) függetlenségi, centráltsági stb. feltevésekkel nem élünk.

A sztochasztikusan kombinált optimalizálási eljárások fenti osztálya a következő lokális konvergenciatulajdonsággal rendelkezik:

3.4. TÉTEL. Az 1.—5. feltételek fennállása esetén a generált $\{\mathbf{x}_k\}$ sorozat 1 valószínűséggel korlátos; az $\{\mathbf{x}_k\}$ sorozat minden határpontja 1 valószínűséggel az f függvénynek egy, a C halmazhoz tartozó lokális minimumhelye.

Tekintsük most a (3.1) feladat (3.24) alakú közelítéseit felhasználó iteratív eljárások globális konvergenciáját. Ebben az esetben a következő feltevésekkel élünk:

1. $f(\mathbf{x})$ folytonos függvény, amely értelmezve van egy, a C determinisztikus megengedett halmazt tartalmazó megfelelő C_0 nyílt halmazon.

2. Megegyezik az előző tétel 2. feltételével.

3. C egy nem-üres, korlátos, összefüggő, nyílt halmaz lezártja.

4. Megegyezik az előző tétel 4. feltételével.

Az 1. és 3. feltételekből következik, hogy (3.1)-nek létezik (nem szükségképpen egyetlen) $\mathbf{x}^*$ globálisan optimális megoldása, legyen $f^*=f(\mathbf{x}^*)$. A globálisan konvergens hibrid eljárásra vonatkozóan a következő feltételt szabjuk:

5. Az $\{\mathbf{x}_k\}$ sorozatot a $\{\mathbf{H}_k\}$ hibrid eljárásokkal határozzuk meg:

$$\mathbf{x}_{k+1} = \mathbf{H}_k(\mathbf{z}_k)$$

$$\mathbf{H}_k\colon\ \alpha_k\mathbf{G}_k(\mathbf{z}_k)+(1-\alpha_k)\mathbf{L}_k(\mathbf{z}_k),\quad 0\leqq\alpha_k\leqq 1,\quad \sum_{k=0}^{\infty}\alpha_k=\infty.$$

Ez a korábban bevezetett jelölésekhez analóg módon azt jelenti, hogy a $\mathbf{H}_k$ eljárás a $\mathbf{G}_k$ és $\mathbf{L}_k$ szubalgoritmusok sztochasztikus kombinációja: az $\mathbf{L}_k$ szubalgoritmusok elvileg tetszőlegesek, míg a $\mathbf{G}_k$-k a következő tulajdonsággal rendelkeznek. Legyen $\varepsilon>0$ esetén $X_{\varepsilon,k}=\{\mathbf{x}\in C+\eta(\mathbf{x}_k)\colon f(\mathbf{x})\leqq f^*+\varepsilon\}$ és tegyük fel, hogy a $\mathbf{G}_k=\mathbf{G}_k(\mathbf{z}_k,\varepsilon)$ szubalgoritmus legalább $p_k(\varepsilon)$ valószínűséggel képes az $X_{\varepsilon,k}$ halmaz valamely pontjának kiválasztására. Itt a $p_k(\varepsilon)$ valószínűségek $\varepsilon>0$ monoton növekvő függvényei, és minden $\varepsilon>0$ és $\{\mathbf{x}_k\}$ sorozat esetén legyen $\liminf_{k\to\infty} p_k(\varepsilon)=p(\varepsilon)>0$. A $\mathbf{G}_k=\mathbf{G}_k(\mathbf{z}_k,\varepsilon_k)$ szubalgoritmusok megválasztása egy, az $\varepsilon_k>0$, $\lim_{k\to\infty}\varepsilon_k=0$ feltételeket kielégítő számsorozatnak megfelelően történik. Az optimum és az optimális megoldás k-adik becslését az

$$f_k=f_{k,\varepsilon_k}\triangleq f(\bar{\mathbf{x}}_k)+\xi(\bar{\mathbf{x}}_k)\triangleq \min_{\substack{0\leqq j\leqq k\\ |\xi(\mathbf{x}_j)|\leqq\varepsilon_k}} [f(\mathbf{x}_j)+\xi(\mathbf{x}_j)]$$

előírás szerint (tehát a k-adik lépésben a legalább ε_k pontosságú függvényértékek alapján) választjuk ki; ez az előírás magában foglalja az aktuális optimumbecslés esetleges pontosítását is.

Érvényes a fenti típusú hibrid optimalizálási eljárások globális konvergenciatulajdonságát kifejező következő állítás.

3.5. TÉTEL. Az 1.—5. feltételek fennállása esetén az $\{\mathbf{x}_k\}$ sorozat 1 valószínűséggel korlátos, $\lim_{k\to\infty} f_k=f^*$ 1 valószínűséggel, és az $\{\bar{\mathbf{x}}_k\}$ optimumbecslések sorozatának minden határpontja 1 valószínűséggel az f függvénynek egy, a C halmazhoz tartozó globális minimumhelye.

Megemlítjük, hogy a 3.4. és 3.5. tételeknek a $\{\beta_k\}$, ill. $\{\alpha_k\}$ sorozatok megválasztására vonatkozó feltételei a 3.4.—3.5. segédtételek értelmében kissé általánosabb alakban is megfogalmazhatók. Megjegyezzük még, hogy a 3.4 és 3.5 tételekhez hasonló konvergenciaeredmények (némileg eltérő feltételek mellett) a 2.1 tétel bizonyításának általánosításával is igazolhatók.

4. Sztochasztikusan kombinált optimalizálási eljárások gépi realizációi

Az előző szakasz 3.4. és 3.5. tételei azt mutatják, hogy a hibrid optimalizálási módszerek lokális, ill. globális konvergenciáját a módszerekben kellő súllyal szereplő $\{\mathbf{R}_k\}$, ill. $\{\mathbf{G}_k\}$ szubalgoritmusok biztosítják, emellett az $\{\mathbf{L}_k\}$ szubalgoritmusok tetszőleges módon választhatók ki. Mivel az $\{\mathbf{R}_k\}$ és $\{\mathbf{G}_k\}$ eljárásokra vonatkozó feltételek is eléggé általánosak, nyilvánvaló, hogy a hibrid módszerek gépi realizációját illetően meglehetősen nagy szabadsággal rendelkezünk. Megengedett

pl., hogy a

$$\mathbf{H}_k\colon\ \alpha_k \mathbf{G}_k + (1-\alpha_k)\mathbf{L}_k$$

hibrid eljárás $\mathbf{G}_k$ és $\mathbf{L}_k$ szubalgoritmusai maguk is néhány további módszer kombinációi legyenek, azaz

$$\mathbf{G}_k\colon \sum_{j=1}^{M_k} \mu_{k_j}\mathbf{G}_{k_j},\quad \mathbf{L}_k\colon \sum_{j=1}^{N_k} \nu_{k_j}\mathbf{L}_{k_j}.$$

Itt $\mu_{k_j} \geqq 0 \sum_{j=1}^{M_k} \mu_{k_j} = 1$, $\nu_{k_j} \geqq 0 \sum_{j=1}^{N_k} \nu_{k_j} = 1$ a $\mathbf{G}_{k_j}$, ill. $\mathbf{L}_{k_j}$ szubalgoritmusok alkalmazásának valószínűségei: a $\mathbf{G}_{k_j}$ módszereknek a $\mathbf{G}_k$ tulajdonságaival kell rendelkezniük, az $\mathbf{L}_{k_j}$ módszerek pedig tetszés szerintiek.

A következőkben illusztrációként egy-egy lokális és globális minimalizálási módszert írunk le, és a velük kapcsolatban végzett numerikus kísérletek eredményeit foglaljuk össze. A lokális módszerre vonatkozó vizsgálatokat korábbi dolgozatunk [48] részletesen tartalmazza, ezért ezekre csak utalunk.

4.1. *Egy lokálisan konvergens hibrid optimalizálási eljárás*

A szóban forgó módszer egy, a (3.1) feladat megoldására irányuló sztochasztikus gradiens típusú adaptív véletlen kereső eljárás, amelyet korábban részletesen vizsgáltunk [48]. Az $\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma_k \mathbf{s}_k$, $k = 0, 1, 2, 3, \ldots$ iterációs lépések végrehajtása során az $\mathbf{s}_k \in E^n$ irányok kiválasztása egy n_d ($1 \leqq n_d \leqq n$) elemű, az n-dimenziós tér egységgömbjének felületén egyenletes eloszlású független vektorokból képezett ortogonális próbairány-halmaz alapján történik. A fenti ortonormált irányok felhasználásával az f függvény (esetleg nem létező) antigradiensének az irányok által kifeszített altérre vonatkozó vetületét is becsüljük. Az így nyert n_d+1 számú irányban végrehajtott adaptív lépéshosszú próbalépések alapján, a legnagyobb függvényérték-redukciót eredményező irányba haladunk tovább. (Nem nehéz belátni, hogy a fenti eljárás alkalmas a 3.4. tétel 5. feltételében leírt irányválasztási stratégia tetszőleges pontosságú közelítésére.) Az $\mathbf{s}_k$ vektor meghatározását követően a γ_k lépéshossz kiválasztása (esetleges $-\mathbf{s}_k$ irányú haladást is megengedő) iránymenti extrapolációval és ezt pontosító kvadratikus interpolációval történik.

A vázolt módszert FORTRAN nyelven programoztuk, a tesztfuttatásokat az *Egyetemi Számítóközpont* R—10 gépén végeztük. A módszer hatékonyságát többféle determinisztikus és sztochasztikus deriváltmentes optimalizálási eljárásával hasonlítottuk össze. Az összehasonlítást minden esetben ismert tesztfeladatok (standardizált kezdőpontból történő) megadott pontosságú megoldásához szükséges (esetünkben véletlen) lépésszámok (átlaga) alapján végeztük. Ezt az indokolja, hogy a vizsgált módszerosztály alkalmazását elsősorban olyan feladatok megoldásánál tartjuk célszerűnek, ahol a (pl. kísérletek végrehajtásával, szimulációval vagy algoritmikus előírással számított) függvényértékek meghatározása az optimalizálási algoritmust szervező műveletekhez képest jelentős időigényű.

A kísérletek során többféle feladatot vizsgáltunk: egyszerű kvadratikus jellegű célfüggvényeket (más véletlen kereső módszerekkel való összehasonlítás során), néhány ismert feltétel nélküli tesztproblémát [21] (determinisztikus gradiensmentes

eljárásokkal való összehasonlítás során), és megoldottunk néhány matematikai programozási tesztfeladatot is (l. [48]). A célfüggvény kiszámításában esetleg fellépő véletlen hibák hatását is modelleztük. E vizsgálatok során az említett determinisztikus tesztfüggvények helyett azok véletlen zajjal torzított variánsait minimalizáltuk: $f(\mathbf{x}) \approx f(\mathbf{x})(1+r)$, ahol r a $[-r_0, r_0]$ intervallumon egyenletes eloszlású, lépésenként függetlenül generált valószínűségi változó.

A számítási eredmények [48, 87] szerint az eljárás a vizsgált egyéb véletlen kereső módszereknél általában hatékonyabbnak, a determinisztikus direkt módszerekkel szintén versenyképesnek bizonyult. Emellett a módszer a fent leírt szerkezetű, 5—10% relatív nagyságot elérő zajhatások mellett is meglehetősen stabilan viselkedett. Ez a tulajdonság különösen fontos sztochasztikus optimalizálási feladatok megoldása során. Megemlítjük még, hogy az eljárást (véletlenszerűen, ill. szakértői javaslatok alapján kiválasztott kezdőpontokból indítva) egy többcélú tározórendszer tervezése során alkalmaztuk: a vizsgált (nem-konvex és ismeretlen differenciálhatósági tulajdonságokkal bíró) modellt, valamint az alkalmazott módszert [50] dolgozatunkban ismertettük.

4.2. *Egy globálisan konvergens hibrid eljárás*

A módszer egy globális és egy lokális kereső algoritmus sztochasztikus kombinációján alapul: ezt az alapeljárást — a hatékonyság növelése érdekében — még bizonyos heurisztikus, többdimenziós kereső technikákkal egészítettük ki.

A globális szubalgoritmus

A szubalgoritmus kiválasztása során az alábbi körülményeket vettük figyelembe. Ismeretes egyrészt, hogy a (3.1) feladat globálisan optimális megoldásának megbízható becslése az $\mathbf{x}_0, \ldots, \mathbf{x}_k$ pontokban végrehajtott célfüggvénykiértékelések esetén csak a feladat szerkezetére vonatkozó megfelelő feltételek mellett lehetséges: egy ilyen feltételrendszert adhatnak pl. a C halmaz korlátosságára és az f függvény *Lipschitz-folytonos* voltára vonatkozó feltevések. Másrészt említettük a *Lipschitz-folytonos* függvények osztályán minimalizáló determinisztikus eljárások bizonyos hátrányait ([15] Vol. 2, 31—48, [95]). Ezért a globális alapeljárást a sztochasztikus módszerek köréből választottuk ki. Amint ezt az utóbbi módszerosztályra vonatkozó elméleti eredmények és számítási tapasztalatok (l. pl. [2, 6, 15, 30, 34, 53, 77, 80, 81, 82, 85, 94]) mutatják, a *Bayes-típusú* (optimális statisztikai döntéseken alapuló) módszerek az „analitikus" jellegű (sztochasztikus approximáció típusú) módszereknél enyhébb feltételek mellett rendelkeznek globális konvergenciatulajdonságokkal, továbbá léteznek eléggé hatékony gépi realizációik.

A *Bayes-típusú* optimalizálási módszerek általános struktúrája a következőképpen írható le [6, 77, 94]. Alapvető feltételezés, hogy a minimalizálandó $f(\mathbf{x})$ függvény valamely $f(\mathbf{x}, \omega)$ sztochasztikus folyamat realizációja ($\mathbf{x} \in C$, az ismeretlen ω paraméter pedig egy $(\Omega, \mathcal{A}, P)$ valószínűségi mező elemi eseményeinek egyike). A P mérték megválasztása lényegileg ekvivalens a minimalizálandó függvények osztályának a priori valószínűségi leírásával. Ezután a $k=0, 1, 2, \ldots$ iterációs lépések során ismertté váló $\mathbf{z}_k = \{\mathbf{x}_i, f(\mathbf{x}_i) = f_i,\ i=0, 1, \ldots, k\}$ keresési információ alapján

adódik a

$$H_{z_k} = H_{x_0, x_1, \ldots, x_k}(f_0, \ldots, f_k) = P(\omega\colon f(\mathbf{x}_i, \omega) < f_i, \quad i = 0, 1, \ldots, k)$$

a posteriori eloszlásfüggvény, amelynek ismeretében kerül kiválasztásra a következő $\mathbf{x}_{k+1} \in C$ mintapont. Valamely S szekvenciális (aktív) döntési stratégia formálisan az

$$S = (X_0, \{D_k\}, \{E_k\}, \{F_k\})$$

alakban állítható elő. Itt

$X_0 \subset C$ az S stratégia rögzített kezdőpontjainak (esetleg üres) halmaza;

D_k: $\{C^{k+1} \cup X_0\} \to C$ az $\mathbf{x}_{k+1}$ mintapont meghatározására vonatkozó döntési szabály;

E_k: $\{C^{k+1} \cup X_0\} \to E^1$ az f függvény minimumhelyére vonatkozó becslés;

F_k: $\{C^{k+1} \cup X_0\} \to \{0, 1\}$ az eljárás befejezési feltétele.

Az S stratégiát egylépéses *Bayes-stratégiának* nevezzük, ha $k = 0, 1, 2, 3, \ldots$ esetén fennáll az

$$\mathbf{x}_{k+1} = D_k(\mathbf{z}_k) = \arg\min_{\mathbf{x} \in C} M\{G[f(\mathbf{x})|\mathbf{z}_k]\} \tag{4.1}$$

reláció, vagyis $\mathbf{x}_{k+1}$ valamely, az f függvénytől és a $\mathbf{z}_k$ feltételtől függő G funkcionál várható értékét minimalizálja. (Az alkalmazások során gyakori pl. $G[f(\mathbf{x})|\mathbf{z}_k] = f(\mathbf{x})|\mathbf{z}_k$ vagy $G[f(\mathbf{x})|\mathbf{z}_k] = [f(\mathbf{x}) - Mf(\mathbf{x})|\mathbf{z}_k]^2$. Bár megfogalmazhatók általánosabb (N-lépéses, ill. végtelen) *Bayes-döntések* is, az optimalizálási gyakorlat szempontjából elsősorban a (4.1) típusú egylépéses stratégiák jöhetnek szóba. Világos ugyanis, hogy a minimalizálandó függvények osztályának a priori modellje általában sok heurisztikus elemet tartalmaz, ezért azt lépésenként szükséges pontosítani. Másrészt — bár az egylépéses *Bayes-döntések* meghatározása az általánosabb döntési szabályokénál egyszerűbb — még a (4.1) típusú feladat is gyakran a (3.1) alapproblémával azonos nehézségi fokú, és csak speciális a priori modellek esetén vezet viszonylag egyszerűen realizálható algoritmushoz.

Amint a *Bayes-típusú* módszerekre vonatkozó numerikus vizsgálatokból kitűnik, a többi globális módszerhez hasonlóan ezeknek is elsősorban az egydimenziós realizációi hatékonyak és már $n = 2, 3$ esetén igen bonyolulttá válhatnak. Ezért a $\mathbf{G}_k$ szubalgoritmust alkalmas irányválasztási stratégia és egy iránymenti globális kereső módszer kombinációjaként hoztuk létre. Az egydimenziós kereső eljárásokra vonatkozó numerikus eredmények összehasonlítása alapján SZTRONGINnak a [94] monográfiában leírt módszerét választottuk ki. Ez az eljárás a *Lipschitz-folytonos* függvények osztályának korlátos véletlen zajjal torzított elemein is globálisan konvergens. Az irányválasztást az n-dimenziós egységgömb felületén egyenletes eloszlású véletlen vektor kiválasztásával hajtjuk végre. (Nem nehéz ellenőrizni, hogy az így kapott $\mathbf{G}_k$ szubalgoritmus folytonos célfüggvény és korlátos megengedett halmaz esetén eleget tesz a 3.5. tétel 5. feltételének.)

A vázolt globális szubalgoritmust különböző heurisztikus módszerekkel egészítettük ki. Ezek alkalmazását az teszi szükségessé, hogy a globális alapalgoritmus numerikus hatékonysága az n dimenziószám növelésével gyorsan csökken, ezért

célszerű a $\mathbf{z}_k$ kereső információ felhasználásával különböző extrapolációs típusú módszereket is alkalmazni. Ezzel kapcsolatban megemlítjük, hogy többdimenziós globális optimalizálási eljárások axiomatikus vizsgálatával foglalkoznak pl. a [81, 82] dolgozatok.

A lokális szubalgoritmus

A sztochasztikusan kombinált eljárás lokális szubalgoritmusát a konjugált gradiens típusú módszerek [20] köréből választottuk ki. Ezt az a már említett körülmény indokolja, hogy a célfüggvény a minimumhely(ek) környezetében gyakran kvadratikus modellel közelíthető és az említett módszerosztály elemei nem csak kvadratikusan konvergensek, hanem általánosabb függvények minimalizálására is hatékonyan alkalmazhatóak (l. pl. [21] numerikus eredményeit). A konjugált gradiens módszerek közül FLETCHER és REEVES [18] algoritmusának egy véges differenciákkal való gradiensközelítéseken alapuló változatát alkalmaztuk.

A hibrid algoritmus

Az eljárás a

(4.2) $$\min_{\mathbf{x}\in C^n} f(\mathbf{x})$$

$$C^n = \{\mathbf{x} = (x_1, \ldots, x_n)\in E^n\colon\ 0 \leqq x_j \leqq 1,\ j = 1, \ldots, n\}$$

speciális globális optimalizálási feladat megoldására irányul. (A C^n megengedett halmaz speciális alakja az algoritmus programozását jelentős mértékben megkönnyítette. Nem nehéz belátni, hogy jóval általánosabb — pl. korlátos döntési változókkal felírt — matematikai programozási feladatok széles osztálya egyszerű transzformáció segítségével a (4.2) alakra hozható.) Feltesszük, hogy (4.2)-nek létezik $\mathbf{x}^*$ optimális megoldása. Az f függvényről csak annyit teszünk fel, hogy értékei C^n pontjaiban valamilyen módszerrel meghatározhatók, eközben a függvénykiértékelések véletlen hibát is tartalmazhatnak.

Az eljárás tetszőleges $\mathbf{x}_0\in C^n$ pontból indul. Adott $\mathbf{x}_0, \ldots, \mathbf{x}_k$ esetén a következő pontot az

$$\mathbf{x}_{k+1} = \mathbf{x}_k^* + \gamma_k \mathbf{s}_k \in C^n, \quad k = 0, 1, 2, \ldots$$

iterációs lépéssel állítjuk elő. Itt $\mathbf{x}_k^* = \arg\min_{0\leqq i\leqq k} f(\mathbf{x}_i)$ (az adott feltételt kielégítő minimális indexű pont kiválasztásával $\mathbf{x}_k^*$ egyértelműen meghatározott). Az $\mathbf{s}_k$ keresési irány és a γ_k lépéshossz kiválasztása a

$$\mathbf{H}_k\colon\ \alpha_k \mathbf{G}_k + (1-\alpha_k)\mathbf{L}_k$$

sztochasztikusan kombinált döntési szabály szerint történik, vagyis α_k, ill. $(1-\alpha_k)$ valószínűséggel a fentebb vázolt véletlen irányú *Sztrongin-féle egydimenziós globális optimalizálás* ($\mathbf{G}_k$), ill. a *Fletcher—Reeves-módszer* egy ciklusa ($\mathbf{L}_k$) kerül végrehajtásra. Az $\{\alpha_k\}$ valószínűségeket az $\alpha_k = \frac{1}{k+1}$ alapelőírással választjuk ki, tehát

a lokális algoritmus végrehajtásának valószínűsége az optimalizálás során fokozatosan növekszik. Ezt az alapszabályt a módszer stabilitásának és hatékonyságának növelése érdekében a következőképpen módosítottuk. Egyrészt a globális szubalgoritmus sikeres végrehajtása után ismét $\mathbf{G}_k$ következik egészen addig, amíg az újabb globális lépések jobb megoldást találnak. Másrészt a lokális szubalgoritmus eredménytelen végrehajtása után a módszer $\mathbf{G}_k$ végrehajtásával folytatódik. (Ezek a módosítások a hibrid eljárás elméleti konvergenciáját nyilvánvalóan megtartják.) A 3.5. tétel 5. feltételének megfelelően a $\mathbf{G}_k$ (és az $\mathbf{L}_k$) szubalgoritmusok végrehajtásának pontosságát az iterációk számának növekedésével fokozatosan növeljük.

Az optimalizálási eljárás akkor fejeződik be, ha a célfüggvény kiértékeléseinek száma, vagy pedig a $\mathbf{G}_k$, ill. $\mathbf{L}_k$ egymást követő sikertelen alkalmazásainak száma egy-egy megadott korlátot elért. (Az iterációs lépés sikeres volt, ha a célfüggvény értéke egy alulról korlátos értékkel csökkent; ez a feltétel természetesen csak az eljárás végességének biztosításához szükséges.) Megemlítjük, hogy a statisztikai döntéselmélet eredményei [6, 77] alapján a globális optimalizálási módszerek megállási szabályai általános alakban leírhatók (l. pl. [2]). Ezek alkalmazását azonban (különösen a többdimenziós esetben) számos tényező nehezíti. Az említett megállási szabályok alkalmazásához ugyanis általában szükséges volna a célfüggvény értékeinek a megengedett tartományon definiálható feltételes eloszlásfüggvényének, vagyis az alábbi χ függvénynek az ismerete:

$$\chi(w) = \chi_{C,f}(w) = P(f(\mathbf{x}) < w | \mathbf{z}_k)$$

(ahol $\mathbf{x}$ a C halmazból egyenletes eloszlás szerint kiválasztott véletlen mintapont.

Miután az optimalizálási módszerek algoritmikusan kiválasztott $\{\mathbf{x}_k\}$ sorozatokat állítanak elő, az ezeknek megfelelő $\{f(\mathbf{x}_k)\}$ értékek halmaza önmagában véve nem szolgáltat a $\chi(w)$ függvény becslésére megbízható alapot. A nehézségeket növeli az $f(\mathbf{x})$ megfigyelt értékeinek esetleges véletlen hibája. Másrészt $\chi(w)$ alapján tulajdonképpen az $f(\mathbf{x})$ extremális értékeit kellene becsülnünk: erre nézve szintén csak heurisztikus jellegű eljárások (l. pl. [13], [15] Vol. 1. 158—165.) ismertek.

Néhány globális optimalizálási tesztfeladat megoldása

A vázolt hibrid algoritmust FORTRAN nyelven programoztuk, a tesztfuttatásokat az *ELTE Számítóközpont* R—10 számítógépén végeztük. A kísérletek eredményeit más globális módszerek eredményeivel vetettük össze: ez az összehasonlítás — a lokális módszerrel kapcsolatos vizsgálatokhoz hasonlóan — ismert tesztfeladatok megoldásához szükséges lépésszámokon alapult. Mivel a globális tesztproblémák esetében nem voltak megadva standard kezdőpontok, a feladatok megoldását véletlenszerűen generált, kezdőpontokból végeztük el. A tesztfeladatokat a [15] könyvben szereplő dolgozatok alapján választottuk ki, ezek általános alakja:

$$\min_{\mathbf{x} \in C} f(\mathbf{x})$$

$$\mathbf{x} = (x_1, \ldots, x_n) \in E^n, \quad C = \{\mathbf{x} \in E^n: \ -\infty < u_j \leqq x_j \leqq v_j < \infty, \ j = 1, \ldots, n\}.$$

Az alább felsorolt tesztfeladatok származására zárójelben utalunk, a továbbiakban

rövidített elnevezésüket is használjuk.

(4.3) GP: $f(x_1, x_2) = [1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2]\times$
$\times[30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)],$
$C = \{-2 \leqq x_j \leqq 2,\ j = 1, 2\}.$ *(Goldstein—Price)*

A GP feladatnak 4 lokális minimumhelye van, globális optimumhelye és optimuma: $\mathbf{x}^*=(0, -1), f^*=3.$

(4.4) B1: $f(x_1, x_2) = \left(x_2-\frac{5{,}1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10,$

$C = \{-5 \leqq x_1 \leqq 10,\ 0 \leqq x_2 \leqq 15\}.$ *(Branin 1. feladata)*

A B1 feladatnak 3 globális optimumhelye van (egyéb lokális optimumok mellett), ezek közelítő értéke és az optimum:

$$\mathbf{x}^*_{(1)} \approx (-3{,}1416;\ 12{,}2750),$$
$$\mathbf{x}^*_{(2)} \approx (\ 3{,}1416;\ 2{,}2750), \quad f^* \approx 0{,}397887.$$
$$\mathbf{x}^*_{(3)} \approx (\ 9{,}4248;\ 2{,}4750),$$

(4.5) B2: $f(x_1, x_2) = 4x_1^2-2{,}1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$

$C = \{|x_1| \leqq 2{,}5;\ |x_2| \leqq 1{,}5\}.$ *(Branin 2. feladata)*

A B2 feladatnak 2 globális és 4 lokális minimuma van, az optimális megoldások és az optimum közelítő értéke:

$$\mathbf{x}^*_{(1)} \approx (\ 0{,}0898;\ -0{,}7126),$$
$$\mathbf{x}^*_{(2)} \approx (-0{,}0898;\ \ 0{,}7126), \quad f^* \approx -1{,}031629.$$

(4.6) S: $f(\mathbf{x}) = f_{m,n}(\mathbf{x}) = -\sum_{i=1}^{m}\frac{1}{\|\mathbf{x}-\mathbf{a}_i\|^2+c_i}$

$\mathbf{x} = (x_1, \ldots, x_n)$

$\mathbf{a}_i = (a_{i1}, \ldots, a_{in}) \in E^n, \quad c_i > 0, \quad i = 1, \ldots, m$ adott konstansok,

$C = \{0 \leqq x_j \leqq 10,\ j = 1, \ldots, n\}.$ *(Shekel-feladatosztály)*

Jelölje S5, S7 és S10 az S feladatosztály konkrét elemeit $n=4$ és a 4.1. táblázat baloldalán feltüntetett paraméterértékek mellett. A feladatok ebben az esetben rendre $m=5, 7, 10$ számú lokális optimumhellyel rendelkeznek: $\mathbf{x}_{(i)} \approx \mathbf{a}_i$. A megfelelő lokális optimumokat a táblázat jobboldala tartalmazza.

A (4.3)—(4.6) feladatok megoldását a következő feltételek mellett végeztük el:

1. A megoldáspont koordinátánkénti előírt pontossága: 10^{-3}.
2. Az optimum meghatározásának előírt pontossága: 10^{-4}.
3. A célfüggvény kiértékeléseinek maximális száma a kétdimenziós tesztfeladatok esetében: 500, a *Shekel-feladatok* esetében pedig $150 \cdot m$, $m=5, 7, 10$.

4.1. TÁBLÁZAT

A feladatok adatai						A feladatok lokális optimumai		
i	$\mathbf{a}_i=(a_{i1}, \ldots, a_{i4})$				c_i	$m=5$	$m=7$	$m=10$
1	4	4	4	4	0,1	−10,1532	−10,4028	−10,5363
2	1	1	1	1	0,2	−5,0552	−5,0877	−5,1285
3	8	8	8	8	0,2	−5,1008	−5,1288	−5,1756
4	6	6	6	6	0,4	−2,6828	−2,7519	−2,8710
5	3	7	3	7	0,4	−2,6304	−2,7659	−2,8066
6	2	9	2	9	0,6		−1,8371	−1,8589
7	5	5	3	3	0,3		−3,7228	−3,8336
8	8	1	8	1	0,7			−1,6753
9	6	2	6	2	0,5			−2,4208
10	7	3,6	7	3,6	0,5			−2,4265

4. A globális és lokális szubalgoritmus egymás utáni sikertelen végrehajtásainak megengedett maximális száma: 5^n, ill. n.

5. Minden feladatot 10 különböző (véletlenül generált) kezdőpontból oldottunk meg, ezek alapján az átlagos lépésszámot határoztuk meg.

6. A tesztfeladatokat véletlen zajhatások figyelembevételével is megoldottuk. A zajt a lépésenként függetlenül generált, 0 várható értékű és $\delta_k=\frac{\delta_0}{2^k}$ szórású r_k normális eloszlású valószínűségi változók felhasználásával, az $f(\mathbf{x}_k)\approx f(\mathbf{x}_k)(1+r_k)$ közelítés alapján modelleztük ($\delta_0\geqq 0$ állandó). Ezzel a modellel a 3.4. és 3.5. tételeknek a véletlen zajhatásokra vonatkozó feltételei tetszőleges pontossággal kielégíthetők.

A következő táblázatban a (4.3)—(4.6) tesztfeladatok megoldására vonatkozó eredményeinket a [15, 72] munkákban ismertetett eredményekkel vetjük össze. A táblázatban az egyes módszercsoportok elnevezéseit az alábbiak szerint rövidítjük:

VEK véletlen irányú egydimenziós globális keresésen alapuló eljárások [9], [15] Vol 2., 1—15.

TM BRANIN trajektória módszere (a stacionárius pontok halmazának algoritmikus elemzése) [8], [15] Vol. 2., 151—163.

ILK Ismételt lokális keresés, véletlenül kiválasztott kezdőpontokból indítva (a módszert interaktív módon végrehajtott cluster-analízissel finomítják) [15] Vol. 2., 49—62.; 151—163., [69].

EFM A célfüggvénynek a megengedett tartományon felvett értékei $\chi(w)$ eloszlásfüggvényének modelljén alapuló módszerek [13], [15] Vol. 1., 158—165; Vol. 2. 85—116.

BM *Bayes-típusú módszerek* (ebben a csoportban szerepelnek a saját eredményeink is) [15] Vol. 2., 117—129., [72].

A 4.2. táblázat eredményeivel kapcsolatban megjegyezzük, hogy csak a saját eredményeink alapszanak véletlen kezdőpontokból indított futtatásokon, továbbá a felsorolt módszerek többségével a tesztfeladatokat csak kisebb pontossággal oldották meg. Az általunk végrehajtott futtatások során a δ_0 zajparaméter értéke 0; 0,1 és 1 volt (a többi módszert csak pontos függvényértékek mellett alkalmazták).

4.2. TÁBLÁZAT

Módszertípus	Tesztfeladatok					
VEK	GP	B1	B2	S5	S7	S10
Bremmerman 1.	210L	250	—	340L	1700L	2500L
Bremmerman 2.	300	160	—	375L	405L	336L
Zilinskas	—	5129	—	L	12121L	8892L
TM						
Gomulka/Branin	—	—	—	5500	5020	4860
ILK						
Törn	2499	1558	—	3679	3606	3874
Gomulka 1.	—	—	—	6654	6084	6144
Gomulka 2.	1495	1318	—	7085	6684	7352
Price	2500	1800	—	3800	4900	4400
EFM						
Fagiuoli	158	1600	—	2514	2519	2518
De Biase—Frontini	378	597	717	620	788	1160
BM						
Mockus	362	189	—	1174	1279	1209
Zilinskas	165	164	—	950L	1017	2224
Pintér 1. $\delta_0=0$	167,LR:1	145	136	371,LR:7	365,LR:5	429,LR:5
2. $\delta_0=0{,}1$	156,LR:4	163,LR:2	157	279,LR:6	228,LR:7	380,LR:5
3. $\delta_0=1$	163,LR:3	153	146	353,LR:7	280,LR:4	407,LR:4

Jelölések: L: lokális optimumhelyhez való konvergencia.
LR: 10 véletlen kezdőpontból indított futtatás során[1]adódott lokális eltérülések száma
—: az adott feladatnak ezzel a módszerrel való megoldására vonatkozóan nincs adatunk.

A táblázat alapján az alábbi következtetések vonhatók le:

1. A VEK módszerek néhány esetben igen gyorsan megoldják a kétdimenziós GP, B1 és B2 feladatokat, az S5, S7, S10 feladatok esetében viszont általában lokálisan eltérülnek (a módszerek nem tudnak a lokális minimumpontokból „megszabadulni").

2. A TM módszerek megbízhatóak, de igen lassúak.

3. Az ILK módszerek szintén lassúak, de — lokális komponensük jóvoltából — a más szerzők által elért eredmények közül a legpontosabbakat adják.

4. Az EFM és BM módszerek összességükben hatékonyabbak, mint az 1.—3. pontokban felsorolt eljárások. Látható, hogy a hibrid algoritmus valamennyi közül a leggyorsabb, ugyanakkor az eredmények pontossága is megfelelő. Meg kell azonban jegyezni, hogy különösen a *Shekel-feladatok* esetében (láthatóan a dimenziószám növekedtével) a módszer hajlamos lokális eltérülésre — a véletlen kezdőpontoktól függően. Ezen kétféle módszerrel is lehet segíteni: egyrészt több véletlen kezdőpontból végrehajtott hibrid optimalizálással, másrészt az eljárás befejezési paramétereinek megváltoztatásával.

Az említett módosítások általában lassítják az eljárást, amely azonban — a fenti numerikus eredmények szerint — még így is várhatóan hatékony marad az egyéb globális módszerekhez képest. Érdemes még megjegyezni, hogy a vizsgált zajhatások a módszer stabilitását és hatékonyságát lényegileg nem csökkentik.

5. Összefoglalás

A dolgozatban optimalizálási feladatok sztochasztikus módszerekkel történő megoldásának kérdéseit vizsgáltuk. A 2. részben megmutattuk, hogy a matematikai programozás determinisztikus, ill. korábban ismert sztochasztikus optimalizálási eljárásainak alkalmazása a gyakorlatban fontos (pl. nem-konvex és(vagy) nem-differenciálható, sztochasztikus szerkezetű) feladatok megoldása során különböző nehézségekbe ütközhet. A sztochasztikus optimalizálási módszereknek a 3. részben ismertetett általános konvergenciatulajdonságai ezek széles körű elvi alkalmazhatóságát biztosítják; numerikus hatékonyságuk növelése érdekében azonban célszerű ezeket megfelelő szerkezetű hibrid optimalizálási eljárások keretében felhasználni. A hibrid módszerek egyszerű gépi realizációira vonatkozó kísérleti eredményeknek más optimalizálási eljárások eredményeivel való összevetése arra utal, hogy ez az általános módszerosztály bonyolult struktúrájú feladatok megoldásának hatékony eszközéül szolgálhat.

IRODALOM

[1] ANIS, A. A. and LLOYD, E. H., "Stochastic reservoir theory: an outline of the state-of-art as understood by applied probabilists", IIASA RR-75-30 (Laxenburg, Austria, 1975).

[2] ARCHETTI, F. and BETRO, B., "A stopping criterion for global optimization algorithms", Scientific Report, (Department of Statistics and Operations Research, University of Pisa, Italy, 1980).

[3] ARROW, K. J., KARLIN, S. and SCARF, H., *Studies in the Mathematical Theory of Inventory and Production*, (Stanford University Press, Stanford, 1958).

[4] ARUNKUMAR, S. and CHON, K., "On optimal regulation policies for certain multi-reservoir systems", *Operations Res.* **26** (1978) 551–562.

[5] BABA, N., "Convergence of a random optimization method for constrained optimization problems", *J. of Opt. Theory and Appl.* **33** (1981) 451–461.

[6] BERGER, J. O., *Statistical Decision Theory. Foundations, Concepts and Methods* (Springer-Verlag, New York, 1980).
[7] BLUM, J. R., "Multidimensional stochastic approximation methods", *Ann. Math. Stat.* **25** (1954) 737–744.
[8] BRANIN, F. H. and HOO, S. K., "A method for finding multiple extrema of a function of N variables", in: *Numerical Methods for Nonlinear Optimization* (Lootsma, F. A., ed.), (Academic Press, London, 1972), 231—237.
[9] BREMMERMAN, H. J., "A method for unconstrained global optimization", *Mathematical Biosciences* **9** (1972) 1–15.
[10] BROOKS, S. H., "Discussion of random methods for locating surface maxima", *Operations Res.* **6** (1958) 244–251.
[11] CHARNES, A. and COOPER, W. W., "Chance-constrained programming", *Management Sci.* **6** (1959) 73–79.
[12] CHARNES, A. and COOPER, W. W., "Deterministic equivalents for optimizing and satisficing under chance constraints", *Operations Res.* **11** (1963) 18–39.
[13] CHICHINADZE, V. K., "The ψ-transform for solving linear and nonlinear programming problems", *Automatica* (1969) 347–355.
[14] DEVROYE, L. P., "Progressive global random search of continuous functions", *Math. Programming* **15** (1978) 330–342.
[15] DIXON, L. C. W. and SZEGŐ, G. P. (eds.), *Towards Global Optimisation, Vol.* 1.-2. (North Holland, Amsterdam, 1975, 1978).
[16] DORFMAN, R., JACOBY, H. D. and THOMAS, H. A. (eds.), *Models for Managing Regional Water Quality* (Harvard University Press, Cambridge, Massachusetts, 1972).
[17] ERMOLIEV, YU. M., "Stochastic quasigradient methods and their application in systems optimization", IIASA WP-82-2 (Laxenburg, Austria, 1981).
[18] FLETCHER, R. and REEVES, C. M., "Function minimization by conjugate gradients", *Comp. J.* **7** (1964) 149–154.
[19] GERENCSÉR, L., Nemlineáris programozási feladatok megoldása szekvenciális módszerekkel. Kandidátusi disszertáció, Budapest, 1975. (MTA SZTAKI Tanulmányok **49** (1976), Budapest.)
[20] HESTENES, M. R., *Conjugate Direction Methods in Optimization* (Springer-Verlag, New York, 1980).
[21] HIMMELBLAU, D. M., "A uniform evaluation of unconstrained optimization techniques", in: *Numerical Methods for Nonlinear Optimization* (Lootsma, F. A., ed.), (Academic Press, London, 1972), 69–97.
[22] HOFFMANN, K. L., "A method for globally minimizing concave functions over convex sets", *Math. Programming* **20** (1981) 22–32.
[23] HOGAN, A. J., MORRIS, J. G. and THOMPSON, H. E., "Decision problems under risk and chance constrained programming: dilemmas in the transition", *Management Sci.* **27** (1981) 698–716.
[24] HORST, R., "An algorithm for nonconvex programming problems", *Math. Programming* **10** (1976) 312–321.
[25] HOWE, C. W., *Benefit–Cost Analysis for Water System Planning* (American Geophysical Union, Water Resources Monograph No. 2., Washington D. C., 1971).
[26] KALL, P., *Stochastic Linear Programming* (Springer-Verlag, Berlin–Heidelberg, 1976).
[27] KALL, P. and PRÉKOPA, A. (eds.), *Recent results in stochastic programming*, Lecture Notes in Economics and Mathematical Systems 179. (Springer-Verlag, New York, 1980).
[28] KELLE, P., Megbízhatósági készletmodellek és alkalmazásuk (MTA SZTAKI Tanulmányok **107** (1980), Budapest).
[29] KIEFER, J. and WOLFOWITZ, J., "Stochastic estimation of the maximum of a regression function", *Ann. Math. Stat.* **23** (1952) 462–466.
[30] KUSHNER, H. J., "A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise", *Trans. ASME ser. D, J. Basic Eng.* **86** (1964) 97–106.
[31] KUSHNER, H. J., "Stochastic approximation algorithms for constrained optimization problems", *Ann. Stat.* **2** (1974) 713–723.
[32] KUSHNER, H. J. and GAVIN, T. L., "Stochastic approximation type methods for constrained systems: Algorithms and numerical results", *IEEE Trans. Aut. Contr.* **AC-19** (1974) 349–357.
[33] KUSHNER, H. J. and KELMANSON, M. L., "Stochastic approximation algorithms of the multiplier type for the sequential Monte Carlo optimization of stochastic systems", *SIAM J. Contr. Opt.* **14** (1976) 827–842.

[34] KUSHNER, H. J. and CLARK, D. S., *Stochastic Approximation Methods for Constrained and Unconstrained Systems* (Springer-Verlag, New York, 1978).
[35] MARKS, D. H., "Models in water resources", in: US Environmental Protection Agency, *A Guide to Models in Governmental Planning and Operations*, Washington D. C., 1975, 103–137.
[36] MARTI, K., "On solutions of stochastic programming problems by descent procedures with stochastic and deterministic directions", *Methods of Operations Research* **33** (1979) 281–293.
[37] MAYER, J., „A STABIL sztochasztikus programozási modellről", *Alk. Mat. Lapok* **2** (1976) 171—187.
[38] MCCORMICK, G. P., "Attempts to calculate global solutions to problems that may have local minima", in: *Numerical Methods for Nonlinear Optimization* (Lootsma, F. A., ed.), (Academic Press, London, 1972), 209–221.
[39] MCMURTRY, G. J. and FU, K. S., "A variable structure automaton used as multimodal searching technique", *IEEE Trans. Aut. Contr.* **AC-11** (1966) 379–387.
[40] MILLER, B. L. and WAGNER, H. M., "Chance-constrained programming with joint constraints", *Operations Res.* **13** (1965) 930–945.
[41] MORAN, P. A. P., *The Theory of Storage* (Methuen, London, 1959).
[42] NASLUND, B. and WHINSTON, A., "A model of multi-period investment under uncertainty", *Management Sci.* **8** (1962) 184–200.
[43] NEMEROW, N. L., *Scientific Stream Pollution Analysis* (McGraw-Hill, New York, 1974).
[44] PINTÉR, J., "A stochastic programming model applied to water resources management", Technical Report No. 11, (Computing Center for Universities, Budapest, 1975).
[45] PINTÉR, J., „Empirikus eloszlásfüggvény-sorozatok maximális eltérésének vizsgálata; alkalmazás egy több periódusú megbízhatósági készletmodellre", *Alk. Mat. Lapok* **1** (1975) 189—195.
[46] PINTÉR, J., „A Sajó-térség vízgazdálkodási modellje", *Vízügyi Közlemények* (1977) 418—426.
[47] PINTÉR, J., "On the maximal distance between two series of empirical distribution functions, with application to an inventory problem", *Methods of Operations Research* **29** (1978) 623–636.
[48] PINTÉR, J., „Véletlen kereső eljárások konvergenciájának és numerikus hatékonyságának vizsgálata", *Alk. Mat. Lapok* **4** (1978) 197—228.
[49] PINTÉR, J., "On the convergence and computational efficiency of random search optimization", *Methods of Operations Research* **33** (1979) 347–362.
[50] PINTÉR, J., "On a stochastic model of reservoir system sizing", in: Proceedings of the 9th IFIP Conference on Optimization Techniques (Iracki, K.–Malanowski, K.–Walukiewicz, S. (eds.)), Lecture Notes in Control and Information Sciences 23. (Springer-Verlag, Berlin, 1980), 546–558.
[51] PINTÉR, J., „Regionális vízminőségvédelmi döntési problémák sztochasztikus modelljei", *Hidrológiai Közlöny* **60** (1980) 364—373.
[52] PINTÉR, J., „Hibrid optimalizálási eljárások nem-differenciálható sztochasztikus feladatok megoldására". *Alk. Mat. Lapok* **7** (1981) 83—97.
[53] POLJAK, B. T., "Nonlinear programming methods in the presence of noise", *Math. Programming* **14** (1978) 87–97.
[54] PRÉKOPA, A., "Reliability equation for an inventory problem and its asymptotic solutions", in: Coll. on Applications of Mathematics to Economics (Akadémiai Kiadó, Budapest, 1965) 317–327.
[55] PRÉKOPA, A., „Sztochasztikus rendszerek optimalizálási problémáiról", doktori értekezés, MTA, Budapest, 1970.
[56] PRÉKOPA, A., "Logarithmic concave measures with application to stochastic programming", *Acta Scientiarium Mathematicarum* **32** (1971) 301–316.
[57] PRÉKOPA, A., "Stochastic programming models for inventory control and water storage problems", in: Coll. Math. Soc. J. Bolyai 7. Inventory Control and Water Storage (Prékopa, A. ed.), (North Holland, Amsterdam, 1973). 229–245.
[58] PRÉKOPA, A., "Contributions to the theory of stochastic programming", *Math. Programming* **4** (1973) 202–221.
[59] PRÉKOPA, A., GANCZER, S., DEÁK, I. and PATYI, K., „A STABIL sztochasztikus programozási modell és annak kísérleti alkalmazása a magyar villamosenergia-iparra", *Alk. Mat. Lapok* **1** (1975) 3—22.

[60] PRÉKOPA, A., RAPCSÁK, T. és ZSUFFA, I., „Egy új módszer sorbakapcsolt tározórendszer tervezésére sztochasztikus programozás felhasználásával", *Alk. Mat. Lapok* **2** (1976) 189—201.
[61] PRÉKOPA, A. és SZÁNTAI, T., „Árvízi tározók méretezése sztochasztikus programozással", *Alk. Mat. Lapok* **2** (1976) 203—217.
[62] PRÉKOPA, A. and SZÁNTAI, T., "On multistage stochastic programming", in: Coll. Math. Soc. J. Bolyai 12. Progress in Operations Research (Prékopa, A. ed.), (North Holland, Amsterdam, 1977), 733–755.
[63] RÉNYI, A., *Valószínűségszámítás* (Tankönyvkiadó, Budapest, 1968).
[64] ROSANOV, YU. A., "Some system approaches to water resources problems: I. Operation under water shortage", IIASA RR-74-17 (Laxenburg, Austria, 1974).
[65] SHUBERT, B. O., "A sequential method seeking the global maximum of a function", *SIAM J. Numer. Anal.* **9** (1972) 397–388.
[66] SIMMONS, D. M., *Nonlinear Programming for Operations Research* (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975).
[67] SOLIS, F. J. and WETS, R. J-B., "Minimization by random search techniques", *Math. of Operations Research* **6** (1981) 19–30.
[68] SZÁNTAI, T., „A Prékopa-féle STABIL sztochasztikus programozási modell numerikus megoldásáról", *Alk. Mat. Lapok* **2** (1976) 93—101.
[69] TÖRN, A. A., "Global optimization as a combination of global and local search", (Åbo Swedish University School of Economics. Technical Report A: 13. Åbo, 1974).
[70] VAN DE PANNE, C. and POPP, W., "Minimum cost cattle feed under probabilistic protein constraints", *Management Sci.* **9** (1963) 405–430.
[71] ZIERMANN, M., „A Szmirnov-tétel alkalmazása egy raktározási problémára", *MTA Mat. Kut. Int. Közl.* **8** (1963) 509—516.
[72] ZILINSKAS, A. G., "On the use of statistical models of multimodal functions for the construction of the optimization algorithms", in: Proceedings of the 9th IFIP Conference on Optimization Techniques (Iracki, K.–Malanowski, K.–Walukiewicz, S. (eds.)), Lecture Notes in Control and Information Sciences 23. (Springer-Verlag, Berlin–Heidelberg, 1980.), 138–147.
[73] Беленький, В. З.—Волконский, В. А.—Иванков, С. А.—Поманский, А. З.—Шапиро, А. Д., Итеративные методы в теории игр и программирования (Наука, Москва, 1974).
[74] Богомолов, Н. А.—Карманов, В. Г., «О сходимости метода случайного поиска для отыскания стационарных точек общей задачи нелинейного программирования», Вестник Московского университета. Серия 15. Вычислительная математика и кибернетика, 1977. № 1. 20—26.
[75] Бочаров, И. Н.—Фельдбаум, А. А., «Автоматический оптимизатор для поиска минимального из нескольких минимумов (глобальный оптимизатор)», Автоматика и Телемеханика, 1962. № 3. 298—301.
[76] Васильев, Ф. П., Численные методы решения экстремальных задач (Наука, Москва, 1980).
[77] Де Гроот, М., Оптимальные статистические решения, (Мир, Москва, 1974).
[78] Денисов, Д. В., «О методе случайного поиска в задачах условной минимизации», Ж. Выч. Мат. и Мат. Физ. **18** (1978) 1103—1111.
[79] Евтушенко, Ю. Г., «Численный метод поиска глобального экстремума (перебор на неравномерной сетке)», Ж. Выч. Мат. и Мат. Физ. **11** (1971) 1390—1403.
[80] Ермольев, Ю. М., Методы стохастического программирования (Наука, Москва, 1976).
[81] Жилинскас, А. Г., «Аксиоматический подход к проблеме экстраполяции в условиях неопределенности», Автоматика и Телемеханика 1979, № 12. 66—70.
[82] Жилинскас, А. Г., «О статистических моделях сложных многоэкстремальных функций и их применении для построения алгоритмов оптимизации», *Problems of Control and Information Theory* **10** (1981) 53–65.
[83] Карманов, В. Г., Математическое программирование (Наука, Москва, 1975).
[84] Колмогоров, А. Н,—Фомин, С. В., Элементы теории функций и функционального анализа (Наука, Москва, 1968).
[85] Моцкус, Й. Б., «Достаточные условия сходимости байесовых методов к абсолютному минимуму непрерывных функций», 67—88, в сборнике: Теория оптимальных решений. вып. 4. Вильнюс, 1978.
[86] Невельсон, М. В.—Хасьминский, Р. З., Стохастическая аппроксимация и рекуррентное оценивание (Наука, Москва, 1972).
[87] Пинтер, Я., «Об одном алгоритме случайного поиска для решения задач безусловной минимизации», Автоматика и Телемеханика 1980, № 12. 76—85.

[88] Пинтер, Я., Математические модели и методы оптимизации в планировании и управлении региональными водохозяйственными системами (обзор), МНИИПУ, Москва, 1980 (принята к печати).
[89] Поляк, Б. Т., «Полунепрерывность интегральных функционалов и теоремы существования в задачах на экстремум», Мат. сборник **120** (1969) № 1.
[90] Поляк, Б. Т., «Сходимость и скорость сходимости итеративных стохастических алгоритмов. 1. Обший случай», Автоматика и Телемеханика, 1976. № 12, 83—94.
[91] Райк, Э., «Качественные исследования в задачах стохастического программирования», Известия АН ЭССР, Сер. физ.-мат., **20** (1971) 8—14.
[92] Райк, Э., «О задачах стохастического программирования с решающими функциями», Известия АН ЭССР, Сер. физ.-мат., **21** (1972) 258—263.
[93] Растригин, Л. А., Системы экстремального управления (Зинатне, Рига, 1974).
[94] Стронгин, Р. Г., Численные методы в многоэкстремальных задачах (Наука, Москва, 1978).
[95] Сухарев, А. Г., «Об оптимальных стратегиях поиска экстремума», Ж. Выч. Мат. и Мат. Физ, **11** (1971) 910—925.
[96] Юдин, Д. Б., Математические методы управления в условиях неполной информации (Советское радио, Москва, 1974).
[97] Юдин, Д. Б., Задачи и методы стохастического программирования (Советское радио, Москва, 1979).

(Beérkezett: 1981. október 28.)

PINTÉR JÁNOS
ELTE SZÁMÍTÓKÖZPONT, OPERÁCIÓKUTATÁSI OSZTÁLY
1093 BUDAPEST, DIMITROV TÉR 8.

STOCHASTIC PROCEDURES FOR SOLVING OPTIMIZATION PROBLEMS

J. PINTÉR

In this paper stochastic optimization methods to solve possibly non-convex and/or non-smooth mathematical programming problems are studied, summarizing also some related previous results [48, 52]. Following the introduction, in Section 2. some practical problems with non-convex and non-differentiable stochastic structure are mentioned. Further on, existence and uniqueness properties of optimal solutions to stochastic programming problems and convergence properties of earlier known iterative stochastic algorithms are summarized. In Section 3. some basic types of convergence characteristics of stochastic optimization procedures are proved, then these results are extended to the case of stochastically combined (hybrid) methods. Finally, in Section 4. some experimental results with hybrid optimization algorithms are presented.

Alkalmazott Matematikai Lapok 7 (1981) 253—269

A KÉMIAI REAKCIÓKINETIKA DIREKT ÉS INVERZ FELADATAIRÓL

TÓTH JÁNOS

Budapest

Bemutatjuk az összetett kémiai reakciók *Volpert-féle* és *Feinberg—Horn—Jackson-féle megadását*, és bebizonyítjuk a kétféle megadás egyenértékűségét. Megfogalmazzuk a reakciókinetika direkt és inverz feladatait, majd megoldjuk ezek közül a következőt: Szükséges és elégséges feltételt adunk arra, hogy egy polinomiális differenciálegyenlet-rendszer egy összetett kémiai reakció indukált kinetikai differenciálegyenlete (=determinisztikus modellje) legyen.

A polinomiális differenciálegyenletek és az összetett kémiai reakciók közötti megfeleltetés egyértelműségét is vizsgáljuk. Végül egy, az indukált kinetikai differenciálegyenletek és a gradiens rendszerek kapcsolatára rámutató állítást mondunk ki.

1. Bevezetés

Előző dolgozatunkban [20] megoldottunk egy inverz feladatot rekeszrendszerek determinisztikus modelljére vonatkozóan. Itt általánosságban megfogalmazzuk a kémiai reakciókinetika direkt és inverz feladatait. Ehhez előkészületként megadjuk az összetett kémiai reakció vagy mechanizmus *Feinberg—Horn—Jackson-féle definícióját*, úgy hogy a [20]-ban szereplő 2.1. definíciót [7] figyelembevételével kissé módosítjuk. Ezután ugyanebben a szellemben megadjuk a *Volpert-féle definíciót* is [23] és [24] alapján. Megmutatjuk, hogy ez a két definíció lényegében egyenértékű. A következő részben általános sémába foglaljuk a formális reakciókinetika problémáit, amelyen belül elhelyezhetők az irodalomban eddig felmerült direkt és inverz feladatok.

Ezután szükséges és elégséges feltételt adunk arra, hogy egy polinomiális differenciálegyenlet-rendszer egy összetett kémiai reakció determinisztikus modellje legyen. E feladat jelentőségéhez a [20] 49. oldalán említetteken kívül megemlítjük, hogy az úgynevezett *elméleti biokémikus* eredményünket a következőképpen hasznosíthatja: valamilyen tudományágból (akár más természettudományból, akár a differenciálegyenletek kvalitatív elméletéből) kiválaszt egy olyan polinomiális differenciálegyenlet-rendszert, amely az általa vizsgált biológiai, vagy biokémiai jelenség szempontjából releváns sajátosságokkal bír. Ezután az alább szereplő 4.2. tétel segítségével megvizsgálja, hogy a differenciálegyenlet-rendszer realizálható-e formális összetett kémiai reakció segítségével, lévén ez szükséges feltétele annak, hogy valóságos összetett kémiai reakcióval realizálható legyen.

Ennek a direkt-inverz feladatpárnak a története így alakult: A feladat direkt részét — tehát az alább szereplő feltétel szükséges voltának megállapítását — rekeszrendszereknél valamivel bővebb osztályra: *elsőrendű reakciókra* HEARON oldotta meg 1953-ban [10, 125. old.], ennyiben tehát [20] 4.1. tétele nem tekinthető újnak. *Másod-*

rendű reakciókra a megoldást KORZUHIN és ZSABOTYINSZKIJ adta meg 1966-ban [25]. Mindezen szerzők folytonossági megfontolásokat használtak, és felhasználták, hogy a kinetikai kezdeti érték problémák megoldása nemnegatív koordinátákkal bír.

Itt bebizonyítjuk azt, hogy a szóban forgó feltétel elégséges is, a direkt-inverz feladatpárt teljes általánosságában oldjuk meg, kizárólag algebrai (sőt, inkább: aritmetikai) eszközöket használunk, és nem használjuk fel a megoldások nemnegativitását.

Ezután vázoljuk a felmerülő problémák néhány csoportját, és néhányat ezek közül meg is oldunk. Többek között megvizsgáljuk, hogy mikor lehet egy indukált kinetikai differenciálegyenlet-rendszer gradiens rendszer.

Eredményeink egy részét a [9] és a [21] előadásban ismertettük.

2. Jelölések és definíciók

Kissé módosított, [7] 1. definíciójához közelebb álló változatát adjuk meg a [20]-beli 2.1. definíciónak.

2.1. DEFINÍCIÓ. *FHJ-összetett kémiai reakciónak* vagy *FHJ-mechanizmusnak* nevezzük az $\langle \mathscr{S}, \mathscr{T}, \mathscr{R}, R\rangle$ objektumot, ahol

(i) $\mathscr{S}$ egy nemüres, véges halmaz, elemeit *kémiai komponenseknek* nevezzük;

(ii) $\mathscr{T} \subset \mathbf{N}_0^{\mathscr{S}}$ egy $N(\in \mathbf{N} \setminus \{1\})$ elemből álló halmaz, elemeit *komplexeknek* hívjuk;

(iii) $\mathscr{R} \subset \mathscr{T} \times \mathscr{T}$ egy reláció, amelyre

(a) $$\forall y \in \mathscr{T} \quad (y, y) \notin \mathscr{R}$$

és

(b) $$\forall y \in \mathscr{T} \exists y' \in \mathscr{T} \quad (y, y') \in \mathscr{R} \vee (y', y) \in \mathscr{R}$$

teljesül, elemei az *elemi reakciók;*

(iv) $R: \mathscr{R} \to (\overline{\mathbf{R}^+})^{(\overline{\mathbf{R}^+})^{\mathscr{S}}}$ egy olyan függvény, amelyik mindegyik elemi reakcióhoz hozzárendeli az elemi reakció sebességét megadó függvényt, ez az *FHJ-kinetika.*

Az FHJ-mechanizmusok halmazát $\mathscr{F}$-fel jelöljük; $(y, y') \in \mathscr{R}$ helyett időnként ezt írjuk: $y' \to y$.

2.2. DEFINÍCIÓ. Az $f = \langle \mathscr{S}, \mathscr{T}, \mathscr{R}, R\rangle \in \mathscr{F}$ FHJ-mechanizmus *FHJ-gráfjának* az $\mathscr{R}$ reláció gráfját hívjuk. Ha $\mathscr{R}$ szimmetrikus, akkor azt mondjuk, hogy az f FHJ-mechanizmus *reverzibilis*, ha pedig $\mathscr{R}$ tranzitív lezártja szimmetrikus, akkor a mechanizmus *gyengén reverzibilis* [7, 89. old.; 11, 109. old.].

2.3. DEFINÍCIÓ. Az $f = \langle \mathscr{S}, \mathscr{T}, \mathscr{R}, R\rangle \in \mathscr{F}$ FHJ-mechanizmus (indukált) *FHJ-kinetikai differenciálegyenlete:*

$$\dot{c}(t) = \sum_{(y', y) \in \mathscr{R}} [R(y', y)](c(t))(y' - y) \quad (t \in \mathscr{D}_c).$$

2.4. DEFINÍCIÓ. Az $f = \langle \mathscr{S}, \mathscr{T}, \mathscr{R}, R\rangle \in \mathscr{F}$ FHJ-mechanizmus R FHJ-kinetikája *tömeghatás típusú,* ha minden $(y', y) \in \mathscr{R}$ elemi reakcióhoz van olyan $K(y', y) \in \mathbf{R}^+$

reakciósebességi állandónak nevezett szám, amellyel

$$R(y', y)(c) = K(y', y)c^y \quad (c \in (\overline{\mathbf{R}^+})^{\mathscr{S}})$$

$(K(y', y) := 0$, ha $(y', y) \notin \mathscr{R})$.

A tömeghatás típusú FHJ-kinetikával rendelkező FHJ-mechanizmusok halmazát $\mathscr{F}_m$-mel fogjuk jelölni, magukat a mechanizmusokat pedig $\langle \mathscr{S}, \mathscr{T}, \mathscr{R}, K \rangle$ alakban fogjuk megadni.

2.1. MEGJEGYZÉS. Az $f = \langle \mathscr{S}, \mathscr{T}, \mathscr{R}, K \rangle \in \mathscr{F}_m$ tömeghatás típusú FHJ-kinetikával rendelkező FHJ-mechanizmus FHJ-kinetikai differenciálegyenlete:

$$\dot{c}(t) = \sum_{(y', y) \in \mathscr{R}} K(y', y)c(t)^y(y' - y) \quad (t \in \mathscr{D}_c). \tag{2.1}$$

2.2. MEGJEGYZÉS. Itteni definícióink abban különböznek a [20]-ban megadottaktól, hogy most nem *sorszámozzuk meg* a komponenseket és a komplexeket. Ez megfelel annak a szemléletnek, amely például a lineáris leképezéseket előnyben részesíti a mátrixokkal szemben.

Definiálunk még néhány, a későbbiekben használt fogalmat.

2.5. DEFINÍCIÓ. Az $f = \langle \mathscr{S}, \mathscr{T}, \mathscr{R}, R \rangle \in \mathscr{F}$ FHJ-mechanizmushoz tartozó *sztöchiometriai tér:*

$$S := \text{span}\,\{y' - y;\ (y, y') \in \mathscr{R}\}.$$

Tetszőleges $D \in (\overline{\mathbf{R}^+})^{\mathscr{S}}$ esetén $(D+S) \cap (\mathbf{R}^+)^{\mathscr{S}}$ *a D-t tartalmazó pozitív reakciószimplex.* Ha létezik olyan $r \in (\mathbf{R}^+)^{\mathscr{S}}$ vektor, amelyik merőleges S-re, akkor a mechanizmus *konzervatív.* Az f mechanizmus *deficienciája:*

$$\delta := N - L - \dim S,$$

ahol L $\mathscr{R}$ gráfja összefüggő komponenseinek (a *láncosztályoknak*) a száma.

Most megadjuk az összetett kémiai reakció VOLPERTtől származó definícióját.

2.6. DEFINÍCIÓ. *V-összetett kémiai reakciónak* vagy *V-mechanizmusnak* nevezzük az $\langle \mathscr{S}, \mathscr{R}, (\alpha, \beta), w \rangle$ objektumot, ahol

(i) $\mathscr{S}$ és $\mathscr{R}$ egy-egy nemüres, véges, egymástól diszjunkt halmaz, elemeiket *kémiai komponenseknek,* illetve elemi reakcióknak hívjuk;

(ii) $(\alpha, \beta) \in (\mathbf{N}_0 \times \mathbf{N}_0)^{\mathscr{S} \times \mathscr{R}}$ egy függvény, értékei *sztöchiometriai együtthatók;*

(iii) $w: \mathscr{R} \to \mathbf{R}^{(\mathbf{R}^{\mathscr{S}})}$ egy olyan függvény, amelyik mindegyik elemi reakcióhoz hozzárendeli az elemi reakció sebességét megadó függvényt, ez a *V-kinetika.*

A V-mechanizmusok halmazát $\mathscr{V}$-vel jelöljük.

2.7. DEFINÍCIÓ. A $v = \langle \mathscr{S}, \mathscr{R}, (\alpha, \beta), w \rangle \in \mathscr{V}$ V-mechanizmus *V-gráfjának* azt az irányított, többszörös élű, hurokél nélküli páros gráfot hívjuk, amelynek két ponthalmaza $\mathscr{S}$ és $\mathscr{R}$; az $\mathscr{S}$ pontjaiból $\mathscr{R}$ pontjaiba, illetve az $\mathscr{R}$ pontjaiból $\mathscr{S}$ pontjaiba vezető irányított élek számát pedig

$$\alpha: \mathscr{S} \times \mathscr{R} \to \mathbf{N}_0 \quad \text{és} \quad \beta: \mathscr{S} \times \mathscr{R} \to \mathbf{N}_0$$

adja meg.

2.8. DEFINÍCIÓ. A $v=\langle \mathscr{S}, \mathscr{R}, (\alpha, \beta), w\rangle \in \mathscr{V}$ V-mechanizmus (indukált) *V-kinetikai differenciálegyenlete:*

$$\dot{c}(t) = \sum_{r\in\mathscr{R}} (\beta(\cdot, r)-\alpha(\cdot, r))w(r)(c(t)) \quad (t\in\mathscr{D}_c).$$

2.9. DEFINÍCIÓ. A $v=\langle \mathscr{S}, \mathscr{R}, (\alpha, \beta), w\rangle \in \mathscr{V}$ V-mechanizmus w V-kinetikája *tömeghatás típusú,* ha minden $r\in\mathscr{R}$-hez létezik olyan (*reakciósebességi állandónak* nevezett) $k(r)\in\mathbf{R}^+$ szám, amellyel minden $\bar{c}\in\mathbf{R}^{\mathscr{S}}$ esetén

$$w(r)(\bar{c}) = k(r)\bar{c}^{\alpha(\cdot, r)}.$$

A tömeghatás típusú V-kinetikával rendelkező V-mechanizmusok halmazát $\mathscr{V}_m$-mel fogjuk jelölni, magukat a mechanizmusokat pedig $\langle \mathscr{S}, \mathscr{R}, (\alpha, \beta), k\rangle$ alakban fogjuk megadni $(k\in(\mathbf{R}^+)^{\mathscr{S}})$.

2.3. MEGJEGYZÉS. A $v=\langle \mathscr{S}, \mathscr{R}, (\alpha, \beta), k\rangle \in \mathscr{V}_m$ tömeghatás típusú V-kinetikával rendelkező V-mechanizmus V-kinetikai differenciálegyenlete:

$$\dot{c}(t) = \sum_{r\in\mathscr{R}} (\beta(\cdot, r)-\alpha(\cdot, r))k(r)\cdot c(t)^{\alpha(\cdot, r)} \quad (t\in\mathscr{D}_c). \tag{2.2}$$

Most pontosan megfogalmazzuk, mit értünk azon, hogy a kétféle megadási mód egyenértékű. Erre azért van szükség, mert mindkét megadási mód elterjedt.

2.1. TÉTEL. Legyen $v=\langle \mathscr{S}, \mathscr{R}, (\alpha, \beta), k\rangle \in \mathscr{V}_m$ tetszőleges, tömeghatás kinetikájú V-mechanizmus. Akkor létezik olyan $f=\langle \mathscr{S}, \mathscr{T}, \mathscr{R}', K\rangle \in \mathscr{F}_m$ tömeghatás kinetikájú FHJ-mechanizmus, amelynek az FHJ-kinetikai differenciálegyenlete megegyezik v V-kinetikai differenciálegyenletével.

Bizonyítás: Megkonstruálunk egy ilyen f-et. Legyen

$$\mathscr{T} := \{\alpha(\cdot, r);\ r\in\mathscr{R}\} \cup \{\beta(\cdot, r);\ r\in\mathscr{R}\},$$

továbbá legyenek az $\mathscr{R}'\subset\mathscr{T}\times\mathscr{T}$ reláció elemei éppen azok az (y', y) párok, amelyekhez létezik olyan $\varrho(y', y)\in\mathscr{R}$ elem, valamint $\alpha(\cdot, \varrho(y', y))$ és $\beta(\cdot, \varrho(y', y))\in$ $\in\mathbf{N}_0^{\mathscr{S}}$ függvény, amelyekkel

$$y = \alpha(\cdot, \varrho(y', y)) \quad \text{és} \quad y' = \beta(\cdot, \varrho(y', y)).$$

Az ilyen $(y', y)\in\mathscr{R}'$ párokra legyen

$$K(y', y) := k(\varrho(y', y)),$$

és legyen K értéke egyébként 0.

Az így előállított $f\in\mathscr{F}_m$ tömeghatás kinetikájú FHJ-mechanizmus FHJ-kinetikai differenciálegyenlete azonos v V-kinetikai differenciálegyenletével, amint azt egyszerű számolás mutatja.

2.4. MEGJEGYZÉS. Bár (2.1) jobboldala csak nemnegatív (koncentráció)-vektorok esetén van értelmezve, (2.2) pedig tetszőleges (koncentráció)-vektorok esetén, az ebből fakadó látszólagos különbségtől eltekinthetünk, mivel ismeretes [24, 355. old.], hogy egy V-kinetikai differenciálegyenlet megoldásai nemnegatív kezdeti feltételek mellett nemnegatívak maradnak.

2.2. TÉTEL. Legyen $f=\langle \mathscr{S}, \mathscr{T}, \mathscr{R}, K\rangle \in \mathscr{F}_m$ tetszőleges, tömeghatás kinetikájú FHJ-mechanizmus. Akkor létezik olyan $v=\langle \mathscr{S}, \mathscr{R}', (\alpha, \beta), k\rangle \in \mathscr{F}_m$ tömeghatás kinetikájú V-mechanizmus, amelynek a V-kinetikai differenciálegyenlete megegyezik f FHJ-kinetikai differenciálegyenletével.

Bizonyítás: Előállítunk egy ilyen v-t. Legyen $\mathscr{R}'$ egy tetszőleges, $\mathscr{S}$-től diszjunkt, $|\mathscr{R}|$ számosságú halmaz, legyen $\varrho: \mathscr{R} \to \mathscr{R}'$ egy kölcsönösen egyértelmű leképezés, és legyen

$$\alpha(\cdot, \varrho(y', y)) := y \quad \text{és} \quad \beta(\cdot, \varrho(y', y)) := y' \quad ((y', y) \in \mathscr{R}).$$

Legyen továbbá

$$k(\varrho(y', y)) := K(y', y).$$

Az így előállított $v \in \mathscr{V}_m$ tömeghatás kinetikájú V-mechanizmus V-kinetikai differenciálegyenlete azonos f FHJ-kinetikai differenciálegyenletével, amint azt egyszerű számolás mutatja.

2.5. MEGJEGYZÉS. Ha φ jelöli a 2.1. tételben leírt, $\mathscr{V}_m$-ről $\mathscr{F}_m$-be vivő leképezést, ψ pedig a 2.2. tételben leírt, $\mathscr{F}_m$-ről $\mathscr{V}_m$-be vivő leképezést, akkor

$$\psi \circ \varphi = id_{\mathscr{V}_m} \quad \text{és} \quad \varphi \circ \psi = id_{\mathscr{F}_m}.$$

Az eddigiek alapján a továbbiakban legtöbbször csak mechanizmusról, kinetikáról, kinetikai differenciálegyenletről (ez utóbbi helyett esetleg *determinisztikus modellről*) fogunk beszélni, és esetenként váltogatjuk a leírási módokat.

2.6. MEGJEGYZÉS. Ha a 2.1. és 2.2. tételben még a reakciók — pontosan megfogalmazható — azonosságát is megköveteljük, akkor φ, illetve ψ már csak a fent megadott lehet.

3. Direkt és inverz feladatok

A reakciókinetika feladatainak túlnyomó többsége megfogalmazható az alábbi általános sémával. Legyen $\mathscr{F}$ a mechanizmusok halmaza, és legyenek A és B nemüres halmazok. Tekintsük a következő diagramot:

$$\mathscr{F} \xrightarrow{\Phi} \mathscr{F}(\mathbf{R}\times A) \xrightarrow{\Psi} B \tag{3.1}$$

($\mathscr{F}(\mathbf{R}\times A) := \{c \subset \mathbf{R}\times A;\ c \text{ függvény}\}$). Itt $f \in \mathscr{F}$ fejezi ki az összetett kémiai reakció vagy mechanizmus *struktúráját*, $\Phi(f)$ írja le a reakcióban végbemenő folyamat *dinamikáját* és Ψ adja meg a folyamatból *származtatott mennyiségeket*.

Egy feladat *direkt feladat*, ha adott $f \in \mathscr{F}$, A, B, Φ és Ψ, és $\Phi(f)$-et, vagy $\Psi(\Phi(f))$-et keressük.

Egy feladat *inverz feladat*, ha A, B, Φ, Ψ és $\Phi(f)$ vagy $\Psi(\Phi(f))$ adott, és f-et keressük, vagy $\mathscr{F}$-nek egy olyan részhalmazát, amelyik tartalmazza f-et.

Direkt feladatot oldunk meg például akkor, amikor kinetikai differenciálegyenletek kvalitatív tulajdonságait határozzuk meg (tudniillik kijelöljük $\mathscr{F}(\mathbf{R}\times A)$ egy olyan részhalmazát, ahová $\Phi(f)$ eshet), vagy amikor egy összetett kémiai reakció sztochasztikus modelljét szimuláljuk [4].

A legfontosabb és legnépszerűbb inverz feladat a *reakciósebességi állandók becslése*. (Ennek egy alkalmazását mutatja be [14].) Ebben az esetben Φ a kinetikai

differenciálegyenlet megoldását a mechanizmushoz hozzárendelő függvény, $\Psi \circ \Phi$ pedig egy megoldáshoz hozzárendeli annak diszkrét időpontokban felvett, hibával terhelt értékeit. Ilyenkor például kijelöljük $\mathscr{F}$-nek egy, azonos gráffal rendelkező mechanizmusokból álló részhalmazát, és ezek közül akarjuk azt kiválasztani, amelyből számítva a kinetikai differenciálegyenlet megoldását, ez a mérési adatoktól a lehető legkevésbé tér el (például négyzetes középben). Mivel a megoldás a sebességi állandóknak nemlineáris függvénye, ezért az általános feladat végleges megoldása numerikus és statisztikai szempontból (a globális optimum meghatározhatatlansága és a becsült paraméterértékek statisztikai jellemzése híján) teljesen reménytelennek tűnik.

Itt elsősorban azzal a direkt-inverz feladatpárral foglalkozunk, amely a kinetikai differenciálegyenletek jellemzését jelenti a polinomiális differenciálegyenletek halmazán belül. Felvetünk, illetve megoldunk néhány ehhez csatlakozó kérdést is.

4. Polinomiális differenciálegyenletek, kinetikai differenciálegyenletek, kinetikai kezdeti érték problémák

4.1. *Polinomiális és kinetikai differenciálegyenletek*

Az alábbiakban $T \in \mathbf{N}_0$ esetén legyen:

$$T^* = \begin{cases} \emptyset, & \text{ha } T = 0, \\ \{1, 2, \ldots, T\}, & \text{ha } T > 0. \end{cases}$$

4.1. DEFINÍCIÓ. Legyen $M, Q \in \mathbf{N}$. Azt mondjuk, hogy a $P: \mathbf{R}^M \to \mathbf{R}^Q$ függvény *(M, Q)-polinom*, ha minden $m \in M^*$, $q \in Q^*$ és $\bar{c}_1, \bar{c}_2, \ldots, \bar{c}_{m-1}, \bar{c}_{m+1}, \ldots, \bar{c}_M \in \mathbf{R}$ esetén a

$$\mathrm{pr}_q \circ P(\bar{c}_1, \bar{c}_2, \ldots, \bar{c}_{m-1}, \cdot, \bar{c}_{m+1}, \ldots, \bar{c}_M): \mathbf{R} \to \mathbf{R}$$

függvény polinom (itt pr_q a q-adik koordinátára való vetítés).

4.1. MEGJEGYZÉS. Ha tehát P (M, Q)-polinom, akkor léteznek olyan

$$I_1, I_2, \ldots, I_Q; \quad J_1, J_2, \ldots, J_Q \in \mathbf{N}_0 \tag{4.1}$$

és

$$\psi_i^q \in \mathbf{R}^+, \quad y_i^q \in \mathbf{N}_0^M \quad (i \in I_q^*, \ q \in Q^*), \tag{4.2}$$

$$\mu_i^q \in \mathbf{R}^+, \quad z_i^q \in \mathbf{N}_0^M \quad (i \in J_q^*, \ q \in Q^*) \tag{4.3}$$

paraméterek, amelyek közül az y_i^q és z_j^q vektorok (amelyeket *kitevővektoroknak* hívnak) páronként különbözők $(i \in I_q^*, j \in J_q^*)$, és amelyekkel minden $\bar{c} \in \mathbf{R}^M$ és $q \in Q^*$ esetén (az üres összeget a szokásnak megfelelően 0-nak véve):

$$\mathrm{pr}_q(P(\bar{c})) = \sum_{i \in I_q^*} \psi_i^q \bar{c}^{y_i^q} - \sum_{i \in J_q^*} \mu_i^q \bar{c}^{z_i^q}.$$

Más szavakkal tehát P minden koordinátája mindegyik változójának egyváltozós polinomja.

4.1. TÉTEL. Tetszőleges $\langle \mathscr{S}, \mathscr{T}, \mathscr{R}, K \rangle$ mechanizmus indukált kinetikai differenciálegyenlete

$$\dot{c} = P \circ c \tag{4.4}$$

alakú, ahol P olyan (M, M)-polinom $\overline{(\mathbf{R}^+)}^M$-re vett megszorítása (vö. a 2.4. megjegyzéssel), amelynek paramétereire a 4.1. megjegyzés jelöléseivel minden $m \in M^*$ és minden $i \in J_m^*$ esetén

$$\mathrm{pr}_m(z_i^m) \in \mathbf{N}$$

teljesül, valahányszor $J_m \in \mathbf{N}$; M pedig $\mathscr{S}$ elemeinek a száma: $M := |\mathscr{S}|$.

Bizonyítás: (2.1)-ből kiindulva kapjuk, hogy

$$(\dot{c}(t))(m) = \sum_{(y', y) \in \mathscr{R}} K(y', y) c(t)^y y'(m) - \sum_{\substack{y \in \mathscr{T} \\ (y', y) \in \mathscr{R}}} y(m) c(t)^y \sum_{y' \in \mathscr{T}} K(y', y).$$

Itt $c(t)^y$ együtthatója a második tagban pontosan akkor különbözik 0-tól, ha $\sum_{y' \in \mathscr{T}} K(y', y) \in \mathbf{R}^+$ és $y(m) \in \mathbf{N}$ egyszerre teljesül.

Most megmutatjuk, hogy ha egy polinomiális differenciálegyenlet P jobb oldala rendelkezik a 4.1. tételben kifejezett tulajdonsággal, azaz a koordinátafüggvények között *nemnegatív kereszthatás* áll fenn, akkor létezik hozzá olyan mechanizmus, amelyiknek éppen ez az indukált kinetikai differenciálegyenlete.

4.2. TÉTEL. Legyen $M \in \mathbf{N}$, és tegyük fel, hogy a P (M, M)-polinom koordinátafüggvényei között nemnegatív kereszthatás áll fenn, és legyenek P paraméterei a (4.1)—(4.3)-ban megadottak. Ekkor létezik olyan mechanizmus, amelynek (4.4) az indukált kinetikai differenciálegyenlete.

4.2. MEGJEGYZÉS. Ezekután a 4.1. és 4.2. tételben szereplő feltételt kielégítő polinomiális differenciálegyenleteket röviden *kinetikai differenciálegyenleteknek,* a megfelelő polinomokat pedig *kinetikai polinomoknak* fogjuk nevezni.

A tétel *bizonyítása:* Megadunk egy olyan $\langle \mathscr{S}, \mathscr{T}, \mathscr{R}, K \rangle$ mechanizmust, amelynek (4.4) az indukált kinetikai differenciálegyenlete. Konstrukciónk egy egyértelműen meghatározott mechanizmushoz vezet, ezt a továbbiakban a (4.4)-hez rendelt *kanonikus mechanizmusnak* nevezzük.

Legyen a szerkesztendő mechanizmusnak M komponense: Legyen $\mathscr{S}$ egy tetszőleges, M elemű halmaz. Legyenek a komplexek:

$$y_i^m,\ y_i^m + e_m \quad (i \in I_m^*,\ m \in M^*);$$

$$z_i^m,\ z_i^m - e_m \quad (i \in J_m^*,\ m \in M^*).$$

(Itt egyes vektorok esetleg többször is szerepelhetnek; e_m a természetes bázis m-edik eleme. $y_i^m \in \mathbf{R}^M$-et azonosítjuk $\mathbf{R}^{\mathscr{S}}$ megfelelő elemével, s ugyanezt tesszük a többi vektorral is.) A nemnegatív kereszthatás miatt $z_i^m - e_m$ minden koordinátája nemnegatív egész szám. Legyenek az elemi reakciók

$$y_i^m \xrightarrow{\psi_i^m} y_i^m + e_m; \quad z_i^m \xrightarrow{\mu_i^m} z_i^m - e_m;$$

ahol a reakciót jelképező nyíl fölé írtuk a megfelelő sebességi állandót a szokással egyezően. Könnyen verifikálható, hogy ennek a mechanizmusnak az indukált kinetikai differenciálegyenlete (4.4), ahol a P polinom paraméterei azonosak a 4.1. megjegyzésben szereplőkkel.

4.3. MEGJEGYZÉS. Tekintsük a következő feltételt: minden $m \in M^*$ és minden $\bar{c}_1, \bar{c}_2, \ldots, \bar{c}_{m-1}, \bar{c}_{m+1}, \ldots, \bar{c}_M \in \overline{\mathbf{R}^+}$ esetén

$$\mathrm{pr}_m \circ P(\bar{c}_1, \bar{c}_2, \ldots, \bar{c}_{m-1}, 0, \bar{c}_{m+1}, \ldots, \bar{c}_M) \geqq 0.$$

Ez a feltétel szükséges, de nem elégséges feltétele a nemnegatív kereszthatás fennállásának, amint az az alábbi példából látható [6]:

$$\dot{c}_1 = c_1 + c_2^2 - 2c_2c_3 + c_3^2$$

$$\dot{c}_2 = 0$$

$$\dot{c}_3 = 0.$$

4.2. *További problémák*

Két, egyéb szempontból is tanulságos példán megmutatjuk, hogy kinetikai differenciálegyenlethez sohasem lehet egyértelműen mechanizmust társítani.

4.1. PÉLDA [6]. A

$$2X \xleftarrow{1} X + Y \xrightarrow{1} 2Y \tag{4.5}$$

mechanizmus tetszőleges mechanizmushoz hozzávehető, anélkül, hogy az utóbbi kinetikai differenciálegyenlete megváltoznék.

(Megjegyzendő viszont, hogy már az az egyszerűbb mechanizmus is, amelyikben Y-t állandó szinten tartott, *külső komponensnek* tekintjük, tehát a

$$2X \xleftarrow{1} X \xrightarrow{1} \emptyset$$

mechanizmus is megkülönböztethető az egyetlen reakciót sem tartalmazó *üres mechanizmustól,* ha *sztochasztikus modelljét* tekintjük. Ekkor ugyanis X darabszámának szórása $+\infty$-hez tart, ha az idő $+\infty$-hez tart.)

4.2. PÉLDA. Tegyük fel, hogy adott az

$$\begin{aligned} \dot{x} &= -2\alpha x^2 v + 2\gamma z^4 \\ \dot{y} &= 3\alpha x^2 v - 3\beta y^3 v^2 \\ \dot{z} &= 4\beta y^3 v^2 - 4\gamma z^4 \\ \dot{v} &= \alpha x^2 v - 2\beta y^3 v^2 + \gamma z^4 \end{aligned} \tag{4.6}$$

differenciálegyenlet. A hozzátartozó kanonikus mechanizmus:

(4.7)

$$
\begin{array}{rcl}
 & \overset{\alpha}{\nearrow} & 2X+2V \\
2X+V & \overset{2\alpha}{\longrightarrow} & X+V \\
 & \underset{3\alpha}{\searrow} & 2X+Y+V \\
 & \overset{2\beta}{\nearrow} & 3Y+V \\
3Y+2V & \overset{3\beta}{\longrightarrow} & 2Y+2V \\
 & \underset{4\beta}{\searrow} & 3Y+Z+2V \\
 & \overset{\gamma}{\nearrow} & 4Z+V \\
4Z & \overset{2\gamma}{\longrightarrow} & X+4Z \\
 & \underset{4\gamma}{\searrow} & 3Z.
\end{array}
$$

Másrészt (4.6) indukált kinetikai differenciálegyenlete a

(4.8)

$$
\begin{array}{ccc}
2X+V & \xrightarrow{\alpha} & 3Y+2V \\
 & {\scriptstyle\gamma}\nwarrow \quad \swarrow{\scriptstyle\beta} & \\
 & 4Z &
\end{array}
$$

mechanizmusnak is. Ez a mechanizmus gyengén reverzibilis, konzervatív és deficienciája 0, ennélfogva a zéró deficiencia tétel (lásd például [19, 287. old.]) szerint minden nemnegatív kezdeti feltétel esetén a (4.6)-ra vonatkozó kezdeti érték probléma megoldása egy korlátos halmazban marad, s így definiálva van a teljes $\overline{\mathbf{R}^+}$ halmazon; létezik egyetlen pozitív egyensúlyi pontja minden pozitív reakciószimplexben, ez az egyensúlyi pont relatíve aszimptotikusan stabilis; és nem létezik (4.6)-nak nemtriviális periodikus megoldása. Ezek az állítások mutatják meg a 4.1.—4.2. tétel jelentőségét a matematikus számára: nem ismerünk ugyanis olyan, nemlineáris differenciálegyenlet-rendszerekre vonatkozó általános tételt, amiből ezek a következtetések egyszerűen levonhatók lennének.

Ez a példa azt mutatja, hogy a kanonikus mechanizmus nem a legegyszerűbb, és nem is minimális semmilyen értelemben. Legfőbb előnye, hogy gyorsan, *algoritmikusan* előállítható.

Az itt felmerülő kérdések néhány csoportja:

(i) Nem túlságosan erős megszorítás-e a nemnegatív kereszthatás fennállása abban az értelemben, hogy egy véletlenszerűen kiválasztott polinomiális differenciálegyenlet úgysem kinetikai? Egyáltalán: milyen sok kinetikai differenciálegyenlet van a polinomiálisok között?

(ii) Az egyszerűbb kezelhetőség kedvéért jó lenne olyan mechanizmust találni, amelyben minimális a komplexek, elemi reakciók, *láncosztályok* (az FHJ-gráf összefüggő komponensei) stb.-k száma. Milyen ésszerű feltevések mellett létezik, illetve egyértelmű egy ilyen minimális mechanizmus?

(iii) Kereshetjük a mechanizmust egy adott — kémiailag fontos — tulajdonsággal rendelkező mechanizmusosztályon belül. Ilyen tulajdonság lehet a konzervativitás, a (gyenge) reverzibilitás, a deficiencia 0 volta stb. Mikor létezik egy adott differenciál-

egyenlethez olyan indukáló mechanizmus, amelyik egy adott osztályba tartozik, mikor egyértelmű ez a mechanizmus, vagy ha nem egyértelmű, akkor mennyire nem?

(Ilyen feladatok megoldásához használhatók [22] eredményei.)

(iv) Hasonló, de bonyolultabb problémákhoz jutunk, ha csak a differenciálegyenletek „lényeges" részét tekintjük.

([20] dolgozatunk (iii)-hoz és (iv)-hez kapcsolódik.)

(v) Kaphatunk-e alkalmas transzformációval nemkinetikai (polinomiális) differenciálegyenletből kinetikait?

Az alábbiakban a fenti felsorolás sorrendjének megfelelően pontosan megfogalmazunk néhány olyan kérdést, amelyre a választ is meg tudjuk adni.

4.3. *Elégséges feltétel nulla deficienciájú indukáló mechanizmus létezésére*

4.2. DEFINÍCIÓ [17]. Az $\langle \mathscr{S}, \mathscr{T}, \mathscr{R}, K \rangle$ mechanizmus *általánosított rekeszrendszer,* ha

(i) minden $\mathscr{A} \in \mathscr{S}$-hez létezik pontosan egy olyan $y \in \mathscr{T}$, hogy

$$\mathscr{A} \in \operatorname{supp} y := \{\mathscr{A}' \in \mathscr{S};\ y(\mathscr{A}') \neq 0\},$$

(ii) minden $y \in \mathscr{T}$ esetén $\operatorname{supp} y$ legfeljebb egyelemű.

4.4. *Megjegyzés.* Más szavakkal a fenti definíció azt jelenti, hogy: minden komponens pontosan egy komplexben szerepel és minden komplex legfeljebb egy komponenst tartalmaz, tehát az elemi reakciók csak az alábbi típusok valamelyikébe tartozhatnak:

$$y^n \mathscr{A}(m) \xrightarrow{k_{im}} y^i \mathscr{A}(i), \tag{4.9}$$

$$y^m \mathscr{A}(m) \xrightarrow{k_{0m}} 0, \quad (\mathscr{A}(m), \mathscr{A}(i) \in \mathscr{S};\ y^m, y^i \in \mathbf{N}), \tag{4.10}$$

$$0 \xrightarrow{k_{m0}} y^m \mathscr{A}(m). \tag{4.11}$$

Ha egy általánosított rekeszrendszerben csak (4.9) típusú elemi reakciók vannak, akkor az *zárt,* ha vannak benne (4.10), illetve (4.11) típusúak is, akkor *szigorúan félig nyílt,* illetve *szigorúan nyílt.*

4.3. TÉTEL [17]. Általánosított rekeszrendszer deficienciája 0.

Bizonyítás: Tegyük fel először, hogy a láncosztályok száma 1. Ekkor a sztöchiometriai tér dimenziója megegyezik egy, az FHJ-gráfot kifeszítő fa éleinek számával, $N-1$-gyel, így tehát $\delta = N-1-(N-1)=0$.

Az általános esetben legyen a láncosztályok száma L, és legyen az l-edik ($l \in L^*$) láncosztályban levő komplexek száma N_l. Ekkor az S sztöchiometriai tér az $S_1, S_2, \ldots, S_L$ alterek direkt összege, ahol az indexezés úgy választható, hogy $\dim S_l = N_l - 1$ álljon. Ennélfogva

$$N = \sum_{l=1}^{L} N_l, \quad \dim S = \sum_{l=1}^{L} \dim S_l$$

és így $\delta = 0$.

4.5. MEGJEGYZÉS. A bizonyításban csak azt használtuk ki, hogy a komplex vektorok — eltekintve az esetleg előforduló 0 vektortól — lineárisan függetlenek, vö. [11, 95. old.]. Ennek egy általánosabb elégséges feltétele az, hogy

$$(y, y') = 0 \quad (\forall\, y \to y' \text{ esetén}), \tag{4.12}$$

(ahol $(\cdot, \cdot)$ most az $\mathbf{R}^{\mathscr{S}}$-beli skalárszorzatot jelöli); (4.12) pedig éppen az *autokatalízis* és az *autoinhibíció* egy speciális fajtájának kizárását jelenti.

4.6. MEGJEGYZÉS. A háromféle típusú általánosított rekeszrendszer indukált kinetikai differenciálegyenlete (2.1) vagy (2.2) specializálásával:

(i) zárt általánosított rekeszrendszeré:

$$\dot{c}_i(t) = \left(-\sum_{j=1}^{M} k_{ij}\right) y^i (c_i(t))^{y^i} + y^i \sum_{j=1}^{M} k_{ji} (c_j(t))^{y^j},$$

(ii) szigorúan félig nyílt általánosított rekeszrendszeré:

$$\dot{c}_i(t) = \left(-\sum_{j=0}^{M} k_{ji}\right) y^i (c_i(t))^{y^i} + y^i \sum_{j=1}^{M} k_{ij} (c_j(t))^{y^j} (\exists i \in M^*:\ k_{0i} > 0),$$

(iii) szigorúan nyílt általánosított rekeszrendszeré:

$$\dot{c}_i(t) = \left(-\sum_{j=0}^{M} k_{ji}\right) y^i (c_i(t))^{y^i} + y^i \sum_{j=1}^{M} k_{ij} (c_j(t))^{y^j} + k_{i0} (\exists i \in M^*:\ k_{i0} > 0),$$

ahol mindvégig:

$$i \in M^*, \quad t \in \overline{\mathbf{R}^+}, \quad k_{ij} \in \overline{\mathbf{R}^+} \quad (i, j \in M^* \cup \{0\}), \quad y^j \in \mathbf{N}.$$

Most megmutatjuk, hogy ha egy kinetikai differenciálegyenlet jobb oldala egyváltozós egytagok összege, akkor — egy kiegészítő feltétel teljesülése esetén — az egyenlethez létezik 0 deficienciájú indukáló mechanizmus (ti.: általánosított rekeszrendszer).

4.4. TÉTEL. Ha a

$$\dot{c}_i(t) = \sum_{j=1}^{M} a_{ij} (c_j(t))^{y^j} + b_i \quad (i \in M^*, \ t \in \overline{\mathbf{R}^+}) \tag{4.13}$$

differenciálegyenlet-rendszerben $i \neq j$, $i, j \in M^*$ esetén

$$y^j \in \mathbf{N}, \quad -a_{ii}, b_i, a_{ij} \in \overline{\mathbf{R}^+},$$

és

$$\beta_j := -\sum_{i=1}^{M} \frac{a_{ij}}{y^i} \geqq 0,$$

akkor létezik olyan általánosított rekeszrendszer, amelynek (4.13) az indukált kinetikai differenciálegyenlete.

Bizonyítás: Megkonstruálunk egy ilyen általánosított rekeszrendszert. Ez az általánosított rekeszrendszer általában különbözni fog a (4.13)-hoz rendelt kanonikus mechanizmustól.

Legyen a szerkesztendő mechanizmusnak M komponense: legyen $\mathscr{S}$ egy tetszőleges M elemű halmaz. Legyenek a komplexek:

$$y^j e_j \quad (j \in M^*)$$

és az üres komplex, $\mathcal{O}$. Legyenek az elemi reakciók:

$$y^j e_j \xrightarrow{a_{ij}/y^i} y^i e_i, \tag{4.14}$$

$$y^j e_j \xrightarrow{\beta_j} \mathcal{O}, \tag{4.15}$$

$$\mathcal{O} \xrightarrow{b_i/y^i} y^i e_i, \tag{4.16}$$

ahol a megfelelő sebességi állandókat a reakciókat jelképező nyíl fölé írtuk. (4.14)—(4.16) természetesen úgy értendő, hogy ha egy elemi reakció sebességi állandója 0, akkor az el is hagyható.

Könnyen verifikálható, hogy ennek a mechanizmusnak az indukált kinetikai differenciálegyenlete éppen (4.13).

4.4. *Egyértelműségi kérdések*

A (4.5) mechanizmus kinetikai differenciálegyenletének jobb oldalán az azonosan 0 (2,2)-polinom áll. Gyengén reverzibilis mechanizmussal ez nem fordulhat elő.

4.4. TÉTEL. Gyengén reverzibilis mechanizmus kinetikai differenciálegyenletének jobboldalán nem állhat az azonosan 0 polinom.

Bizonyítás: Legyen az $\langle \mathscr{S}, \mathscr{T}, \mathscr{R}, K \rangle$ mechanizmus sztöchiometriai altere S, és legyen e mechanizmus indukált kinetikai differenciálegyenlete

$$\dot{c} = P \circ c.$$

Akkor [7, 90. old.]

$$S' := \operatorname{span}(\operatorname{Im} P) = S,$$

viszont $\dim S \geqq 1$ miatt S' nem állhat csak a 0 vektorból.

4.3. PÉLDA. A

$$2X \underset{3}{\overset{3}{\rightleftharpoons}} X+Y \underset{2}{\overset{2}{\rightleftharpoons}} 2Y$$

és az

$$X+Y \underset{1}{\overset{1}{\rightleftharpoons}} 2X \underset{1}{\overset{1}{\rightleftharpoons}} 2Y$$

mechanizmus indukált kinetikai differenciálegyenlete:

$$\dot{x} = -\dot{y} = -3x^2+xy+2y^2.$$

Ez a példa tehát azt mutatja, hogy az viszont nem igaz, hogy egy kinetikai differenciálegyenletet csak *egy* mechanizmus indukálhat (akár a reverzibilis és konzervatív reakciók osztályában is). A reverzibilitást további tulajdonságokkal kiegészítve már kaphatunk elégséges feltételt az egyértelműségre [13].

A következőkben azt vizsgáljuk, hogy milyen módon lehet, illetve milyen módon nem lehet egy polinomiális, nemkinetikai differenciálegyenletből kinetikait kapni.

4.5. *Polinomiális differenciálegyenletek beágyazása kinetikai differenciálegyenletekbe*

Mindig elérhető, hogy a megoldás koordinátafüggvényeinek számát — eggyel — növelve olyan függvényt kapjunk, amely már kinetikai differenciálegyenletnek tesz eleget.

4.5. TÉTEL. Tetszőleges nem-kinetikai P (M, M)-polinomhoz és tetszőleges $D \in (\overline{\mathbf{R}^+})^M$-hez létezik olyan Q $(M+1, M+1)$-polinom, hogy a

$$\dot{c} = P \circ c$$

$$c(0) = D$$

kezdeti érték probléma megoldását egy $d \in \mathscr{F}(\mathbf{R} \times \mathbf{R})$ függvénnyel kiegészítve:

az $x \in \mathscr{F}(\mathbf{R} \times \mathbf{R}^{M+1})$ függvény az

$$x := (c, d)$$

$$\dot{x} = Q \circ x \tag{4.17}$$

$$x(0) = x_0$$

kinetikai differenciálegyenletre vonatkozó kezdeti érték probléma megoldása.

4.7. MEGJEGYZÉS. Ha *kinetikai kezdeti érték problémának* egy kinetikai differenciálegyenletre vonatkozó, nemnegatív kezdeti feltételhez tartozó kezdeti érték problémát nevezünk, akkor ezen szóhasználattal élve a tétel bizonyításában megkonstruálandó kezdeti érték probléma nemkinetikai lesz. Annál inkább megvizsgálandó a későbbiekben, hogy milyen következtetések vonhatók le mégis a tételből.

A tétel *bizonyítása:* Legyen $d: \mathscr{D}_c \to \mathbf{R}$ a következő függvény: $\dot{d} = 0$, $d(0) = -1$. Minden $m \in M^*$ és minden $P_m(\bar{c})$-ban összeadandóként szereplő $-\alpha \bar{c}^y$ $(\alpha \in \mathbf{R}^+$, $\bar{c} \in \mathbf{R}^M$, $y \in \mathbf{N}^M$, $pr_m y = 0)$ tag helyett írjunk $\alpha \bar{c}^y d^1$-et, a többi kitevővektort pedig a 0 szám hozzávételével egészítsük ki. Az így definiált Q polinomban már nincs negatív kereszthatás, és $x := (c, d)$ valóban az

$$\dot{x} = Q \circ x$$

$$x(0) = x_0 := (D, -1)$$

kezdeti érték problémának tesz eleget.

4.4. PÉLDA. Legyen $\alpha, \beta, \gamma, \delta \in \mathbf{R}^+$; $e_0, \delta_0, c_0, p_0 \in \mathbf{R}^+$. Akkor az

$$\dot{e} = (\alpha - \delta) ep - \alpha e^2 - (\beta + \gamma + \alpha p_0 + \alpha s_0 - \alpha e_0) e + (\beta + \gamma)(e_0 + c_0)$$

$$\dot{p} = -\delta ep \underline{\boxed{-\gamma e}} + \gamma (e_0 + c_0) \tag{4.18}$$

$$e(0) = e_0 \quad p(0) = p_0$$

nem-kinetikai differenciálegyenletre vonatkozó kezdeti érték problémához a 4.5. tétel az

$$\dot{e} = (\alpha-\delta)ep-\alpha e^2-(\beta+\gamma+\alpha p_0+\alpha s_0)e+(\beta+\gamma)(e_0+c_0)$$
$$\dot{p} = -\delta ep+\gamma ed+\gamma(e_0+c_0)$$
$$\dot{d} = 0$$
$$e(0) = e_0, \quad p(0) = p_0, \quad d(0) = -1$$

kinetikai differenciálegyenletre vonatkozó nem-kinetikai kezdeti érték problémát rendeli. (A (4.18) képletben bekeretezett $-\gamma e$ tag negatív kereszthatást fejez ki.) Másrészt viszont (4.18) megoldásai azonosak az

$$\dot{e} = -\alpha es+\beta c+\gamma c-\delta ep$$
$$\dot{s} = -\alpha es+\beta c$$
(4.19)
$$\dot{c} = \alpha es-\beta c-\gamma c+\delta ep$$
$$\dot{p} = \gamma c-\delta ep$$
$$e(0) = e_0, \quad s(0) = s_0, \quad c(0) = c_0, \quad p(0) = p_0$$

kinetikai kezdeti érték probléma megoldásának az első két koordinátára való vetületével. Felmerül tehát az a — pillanatnyilag nyílt — kérdés, hogy mikor, milyen feltételek mellett, és milyen algoritmussal ágyazható be egy nemkinetikai polinomiális differenciálegyenletre vonatkozó kezdeti érték probléma megoldása egy kinetikai kezdeti érték probléma megoldásába. ((4.19) a jól ismert *Michaelis—Menten-reakció* kinetikai differenciálegyenlete.)

4.6. *Kinetikai gradiens rendszerek*

Végezetül a 4.1. tétel alkalmazásaként megvizsgáljuk, hogy mi a szerepük a *gradiens rendszereknek* a kémiai reakciókinetikában. Ez a feladat a (3.1) diagram nyelvén úgy fogalmazható meg, hogy Φ $\mathscr{F}$-nek milyen részhalmazát képezi le $\mathscr{F}(\mathbf{R}\times A)$ egy adott részhalmazára.

Matematikai szempontból ez azért érdekes, mert a gradiens rendszerek azok, amelyek könnyen tanulmányozhatók az egzotikusnak nevezett jelenségek egy csoportja leírására szolgáló *katasztrófaelméleten* belül [5, 246. old.; 16, 55. old.].

Termodinamikai szempontból a kérdésnek az a jelentősége, hogy egyes iskolák szerint [3, 8] csak gradiens rendszerrel leírható jelenségeket érdemes tanulmányozni. Kérdés tehát, hogy az ilyen jelenségek osztályába milyen fajta kémiai reakciók tartoznak.

4.3. Definíció. A

$$\dot{c} = P\circ c$$

differenciálegyenletet $(P\in\mathscr{F}(\mathbf{R}^M\times\mathbf{R}^M);\ M\in\mathbf{N})$ *gradiens rendszernek* nevezzük, ha létezik olyan $V\in\mathscr{C}^1(\mathscr{D}_p, \mathbf{R})$ függvény, amellyel

$$P = V'(= \operatorname{grad} V)$$

teljesül. V szokásos elnevezése: *potenciál.*

További előkészítő definíciókra van szükségünk.

4.4. DEFINÍCIÓ. Az $\langle \mathscr{S}, \mathscr{T}, \mathscr{R}, K \rangle$ mechanizmus *keresztkatalitikus,* ha minden $y \to y'$ elemi reakció esetén

(i) vagy $y' - y \in \mathbf{N}_0^{\mathscr{S}}$,

(ii) vagy az y reaktáns komplex $\mathbf{R}^{\mathscr{S}}$ valamelyik báziselemének n-szerese valamilyen $n \in \mathbf{N}_0$-lal.

4.8. *Megjegyzés.* Egy keresztkatalitikus mechanizmusban tehát nincsen olyan kémiai komponens, amelyik egy másik komponens fogyását okozná. Semmit nem tettünk fel a komponensek saját magukra gyakorolt hatásáról. — Definíciónk — legalább is szellemében — összhangban van a kémiai irodalomban elfogadott elnevezésekkel, vö. [2].

4.5. DEFINÍCIÓ. Egy mechanizmus *kanonikusan keresztkatalitikus,* ha az általa indukált kinetikai differenciálegyenlethez tartozó kanonikus mechanizmus keresztkatalitikus.

4.9. MEGJEGYZÉS. Ha egy mechanizmus keresztkatalitikus, akkor kanonikusan is keresztkatalitikus, az állítás megfordítása pedig nyilván nem igaz.

4.6. DEFINÍCIÓ. Egy mechanizmus *gyengén reális,* ha minden $y \to y' \in \mathscr{R}$ elemi reakció esetén

(i) vagy $\mathscr{R}_y \subset \{0, 1\}$,

(ii) vagy az y reaktáns komplex $\mathbf{R}^{\mathscr{S}}$ valamelyik báziselemének n-szerese valamilyen $n \in \mathbf{N}_0$-lal.

4.10. MEGJEGYZÉS. Nyilvánvaló, hogy

(i) az általánosított rekeszrendszerek gyengén reálisak, és

(ii) a reális mechanizmusok — amelyek reaktáns komplexe nem hosszabb kettőnél [19, 246. old.] — egyben gyengén reálisak is.

4.6. TÉTEL. Ha egy gyengén reális mechanizmus indukált kinetikai differenciálegyenlete gradiens rendszer, akkor a mechanizmus kanonikusan keresztkatalitikus.

Bizonyítás: Legyen $\langle \mathscr{S}, \mathscr{T}, \mathscr{R}, K \rangle$ egy olyan gyengén reális mechanizmus, amelynek indukált kinetikai differenciálegyenlete gradiens rendszer és legyen $|\mathscr{S}| =: M$. Az állítással ellentétben tegyük fel, hogy létezik olyan $m, m' \in M^*$; $m \neq m'$ és $k \in \mathbf{R}^+$, hogy a mechanizmus által indukált

$$\dot{c} = P \circ c$$

kinetikai differenciálegyenlet m-edik sorának jobb oldala tartalmaz egy $-kc_m c_{m'} \cdot \pi$ alakú tagot, ahol π c bizonyos, c_m-től és $c_{m'}$-től különböző indexű koordinátái hatványainak — esetleg üres — szorzata. Ismeretes [15, 10.35 Megjegyzés], hogy — ha $P = (P_1, P_2, \ldots, P_M)$ — P-hez pontosan akkor létezik potenciál, ha

$$\partial_{m'} P_m = \partial_m P_{m'} \quad (m, m' \in M^*).$$

Esetünkben

$$\partial_{m'} P_m(c) = -kc_m \pi + \ldots = \partial_m P_{m'}(c)$$

kellene, hogy teljesüljön, azaz $P_{m'}(c)$-ben kellene, hogy egy $-kc_m^2 \pi/2$ alakú tag álljon. Ez viszont ellentmond a 4.1. tétel állításának.

4.11. MEGJEGYZÉS. Ha egy keresztkatalitikus mechanizmus elemi reakciói között (i) típusú (4.4. definíció) elemi reakciók is szerepelnek, akkor az nem konzervatív. Így tehát, ha egy konzervatív mechanizmus indukált kinetikai differenciálegyenlete gradiens rendszer, akkor a mechanizmus csak az alábbi két osztály egyikébe eshet:

(i) a nem gyengén reális és nem kanonikusan keresztkatalitikusok közé, vagy

(ii) az olyan kanonikusan keresztkatalitikusok közé, amelyek nem tartalmaznak (i) típusú (4.4. definíció) elemi reakciókat.

4.12. MEGJEGYZÉS. A (ii) típusba (4.10. megjegyzés) tartoznak a szimmetrikus K mátrixszal bíró zárt rekeszrendszerek. Ez az állítás azért ellentétes az [1] 165. oldalán szereplővel, mert definícióink különböznek. [1] második példája viszont egy konzervatív, nem gyengén reális és nem keresztkatalitikus mechanizmus, amelynek indukált kinetikai differenciálegyenlete nem gradiens rendszer a mi definíciónk értelben, így a példa állításainkkal összhangban van.

Köszönettel tartozom HÁRS VERÁnak — akivel közösen kezdtünk foglalkozni ezzel a témakörrel — a kézirat különböző változataihoz fűzött megjegyzéseiért, valamint ÉRDI PÉTERnek és PÓTA GYÖRGYnek az irodalom összeállításához nyújtott segítségéért.

IRODALOM

[1] BATAILLE, D. E. B., EDELEN, J. and KESTIN, J., "Nonequilibrium thermodynamics of the nonlinear equations of chemical kinetics", *J. Non-Equilib. Thermodyn.* **3** (1978) 153–168.
[2] BAZSA, GY. és BECK, M., „Autokatalízis—autoinhibíció, sajátkatalízis—sajátinhibíció: a reakciótermékek és a reaktánsok specifikus kinetikai hatásai", *Kém. Közl.* **36** (1971) 167–183.
[3] EDELEN, J., "Asymptotic stability, Onsager fluxes and reaction kinetics", *Int. J. Engng. Sci.* **11** (1973) 819–839.
[4] ÉRDI, P., SIPOS, T. és TÓTH, J., „Összetett kémiai reakciók sztochasztikus szimulálása számítógéppel", *Magy. Kém. Foly.* **79** (1973) 97–108.
[5] FARKAS, M., „Folyamatok kvalitatív vizsgálatáról", *Alk. Mat. Lapok* **2** (1976) 237–257.
[6] FEINBERG, M. (Személyes közlés, Heidelberg, 1980.)
[7] FEINBERG, M. and HORN, F. J. M., "Chemical mechanism structure and the coincidence of the stoichiometric and kinetic subspaces", *Arch. Ratl. Mech. Anal.* **66** (1977) 83–97.
[8] GYARMATI, I., "On the phenomenological basis of irreversible thermodynamics, I. (Onsager's theory)", *Per. Polytechn.* **5** (1961) 219–243.
[9] HÁRS, V. and TÓTH, J., "On the inverse problem of reaction kinetics" in: *Qual. Theory Diff. Eqs.* Ed. L. Hatvani, Coll. Math. Soc. J. Bolyai 30, 363—379.
[10] HEARON, J. Z., "The kinetics of linear systems with special reference to periodic reactions", *Bull. Math. Biophys.* **15** (1953) 121–141.
[11] HORN, F. and JACKSON, R., "General mass action kinetics", *Arch. Ratl. Mech. Anal.* **47** (1972) 81–116.
[12] LORENZ, P., "Deterministic nonperiodic flow", *J. Atmospheric Sci.* **20** (1963) 130–141.
[13] NEMES, I., *„Összetett kémiai folyamatok kinetikájának és mechanizmusának néhány problémájáról"* (Kandidátusi értekezés, Budapest, 1973.)
[14] RÁCZ, I., GYARMATI, L. és TÓTH, J., „Hidrofil és lipofil karakterű felületaktív anyagok befolyása a szalicilsavtranszport kinetikájára háromfolyadékteres modell esetén", *Acta Pharmaceutica Hung.* **47** (1977) 201–208.
[15] RUDIN, W., *A matematikai analízis alapjai* (Műszaki Könyvkiadó, Budapest, 1978.)
[16] THOM, R., *Structural Stability and Morphogenesis* (W. A. Benjamin, Inc., Reading, Ma., 1975).
[17] TÓTH, J., "What is essential to exotic behaviour?", *React. Kinet. Catal. Lett.* **9** (1978) 377–381.
[18] TÓTH, J., "Gradient systems are cross-catalytic", *React. Kinet. Catal. Lett.* **12** (1979) 253–257.
[19] TÓTH, J. és ÉRDI, P., „A formális reakciókinetika modelljei, problémái és alkalmazásai", *A kémia újabb eredményei* **41** (1978) 226–352.
[20] TÓTH, J. és HÁRS, V., „A rekeszrendszerek inverz feladatáról", *Alk. Mat. Lapok* **5** (1979) 49–61.

[21] TÓTH, J. and HÁRS, V., "The inverse problems of chemical reaction kinetics", Lecture presented at the *Workshop on Modelling of Chemical Reaction Systems* (Heidelberg, Sept. 1–5, 1980).
[22] WILLAMOWSKI, K.-D. and RÖSSLER, O. E., "Contributions to the theory of mass action kinetics, I. Enumeration of second order mass action kinetics", *Z. Naturforsch.* **33a** (1978) 827–833.
[23] Вольперт, А. И., «Дифференциальные уравнения на градах», *Мат. сборник* **88** (1972) 578—588.
[24] Вольперт, А. И., Худяев, С. И., *Анализ в классах разрывных функций и уравнения математической физики* (Наука, Москва, 1975).
[25] Жаботинский, А. М., Корзухин, М. Д., «Математическое моделирование кинетики гомогенных химических систем, I—III.» in: *Колебательные процессы в биологических и химических система*, Рег, Г. М. Франк, А. М. Жаботинский, А. М. Молчанов, Д. С. Чернавский и С. Э. Шноль (Изд. Наука, Москва, 1967.) 223–231. 231–242, 242–251.

(Beérkezett: 1981. január 13.)

TÓTH JÁNOS
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1111 BUDAPEST, KENDE U. 13—17.

ON DIRECT AND INVERSE PROBLEMS OF CHEMICAL REACTION KINETICS

J. TÓTH

The definition of complex chemical reactions or mechanisms by VOLPERT and by FEINBERG, HORN and JACKSON are presented and they are shown to be equivalent. The direct and inverse problems of reaction kinetics are formulated. A necessary and sufficient condition is given for a polynomial system of differential equations to be the induced kinetic differential equation (i.e. the deterministic model) of a complex chemical reaction.

Related questions pertaining mainly to uniqueness and the significance of gradient systems in reaction kinetics are investigated as well.

Alkalmazott Matematikai Lapok 7 (1981) 271—278

A KRIGING-MÓDSZER NÉHÁNY TULAJDONSÁGA

MOLNÁR SÁNDOR SIDNEY YAKOWITZ SZIDAROVSZKY FERENC

Budapest

A dolgozatban az univerzális és a klasszikus *Kriging-módszer* néhány matematikai tulajdonságát vizsgáltuk meg. Alkalmas feltételek mellett kimutattuk a módszerek konvergenciáját, ha az alappontok számát minden határon túl növeljük. Megvizsgáltuk a pontatlan függvényértékek hatását is.

1. Bevezetés

A geostatisztikai gyakorlatban ma már alapvető fontosságúak a *Kriging-módszer* ([4]) változatai. Mint tudjuk, segítségükkel az ásványvagyon mennyiségi és minőségi paramétereit tudjuk megbecsülni ([1], [2], [3]). Ez a becslés kétféle típusú lehet: egyrészt egy vagy több paraméter alappontoktól (azaz fúrási helyektől) különböző helyeken felvett értékeit, vagy valamely tartományon (blokkon, telepen) felvett átlagértékét becsülhetjük meg. A módszer nemcsak a paraméterértékeket, illetve átlagértékeket adja meg, hanem a becslés szórását, azaz a becslés bizonytalanságának a mértékét is szolgáltatja. Ezzel az ásványvagyon kategóriákba való sorolását is nagymértékben elősegíti ([3]).

Jelen dolgozatban a *Kriging-módszer* általánosított változatát, az ún. univerzális *Kriging-módszert* ([3]) vizsgáljuk meg. Alkalmas feltételek fennállása esetén a módszer konvergenciáját mutatjuk ki, valamint konkrét hibaformulákat is levezetünk.

2. Az univerzális Kriging-módszer

Legyen $f(\mathbf{x})$ valós értékű függvény, amely az $\mathbf{x}$ helyen valamely mennyiségi vagy minőségi paraméter értékét jelöli. Tegyük fel, hogy az $\mathbf{x}_1, \ldots, \mathbf{x}_N$ helyeken ismerjük az $f(\mathbf{x}_i)$ közelítő értékeket, azaz ismertek az

$$y_i = f(\mathbf{x}_i) + n(\mathbf{x}_i) \quad (1 \leqq i \leqq N)$$

értékek, ahol az $n(\mathbf{x}_i)$ hibatagokról feltesszük, hogy zérus várható értékűek, azonos $\sigma > 0$ szórásúak és egymástól függetlenek. A *Kriging-módszer* a következő kérdésekre keres választ ([3]):

1. *Feladat.* Legyen $\mathbf{x}^*$ a mérési helyektől különböző pont. Becsüljük meg az $f(\mathbf{x}^*)$ függvényértéket és határozzuk meg a becslési hibát jellemző

$$E[(f_N(\mathbf{x}^*) - f(\mathbf{x}^*))^2]$$

varianciát, ahol $f_N(\mathbf{x}^*)$ adja a becsült függvényértéket.

2. *Feladat.* Legyen D valamilyen tartomány (blokk, telep). Becsüljük meg az

$$\frac{1}{|D|}\int_D f(\mathbf{x})\,d\mathbf{x}$$

átlagértéket, és határozzuk meg a becslési hiba varianciáját. Itt $|D|$ a D tartomány mértékét (területét, térfogatát) jelöli.

Megjegyezzük, hogy az 1. feladat a vizsgált paraméter valamely helyen való értékének becslésére vonatkozik, míg a 2. feladattal a D tartományon való átlagértékét adhatjuk meg. Ha $f(\mathbf{x})$ az $\mathbf{x}$ helyen való telepvastagságot adja meg, akkor az átlagértéket a $|D|$ mennyiséggel beszorozva a D tartományhoz tartozó ásványvagyon mennyiségét kapjuk meg.

Az $f(\mathbf{x})$ függvényről általában feltesszük a következőket ([4]):

$$E[f(\mathbf{x})] = \sum_{j=1}^{J} a_j \varphi_j(\mathbf{x}), \tag{2.1}$$

ahol a $\varphi_j(\mathbf{x})$ függvények adottak, az a_j együtthatók ismeretlenek, valamint

$$\frac{1}{2}\operatorname{Var}[f(\mathbf{x}+\mathbf{h})-f(\mathbf{x})] = \gamma(\mathbf{h}), \tag{2.2}$$

ahol Var a variancia jele. E tulajdonság szerint a (2.2) varianciaérték csak a $\mathbf{h}$ megváltozásától függ, $\mathbf{x}$-től független.

A $\gamma(\mathbf{h})$ ún. variogram függvényt tapasztalati varianciaként az adott függvényértékekből könnyen megbecsülhetjük konkrét $\mathbf{h}$ értékek esetén, majd az így véges sok pontban ismert függvényt a legkisebb négyzetek módszerével simítják. A legfontosabb variogram-függvény típusok a következők ([2]):

$$\gamma(\mathbf{h}) = \omega|\mathbf{h}|^{\alpha} \quad (0 < \alpha < 2) \qquad \text{(polinomiális)}$$

$$\gamma(\mathbf{h}) = \omega(1-e^{-\alpha|\mathbf{h}|}) \qquad \text{(exponenciális)}$$

$$\gamma(\mathbf{h}) = \omega(1-e^{-\alpha|\mathbf{h}|^2}) \qquad \text{(Gauss-típusú)}$$

$$\gamma(\mathbf{h}) = \begin{cases} \dfrac{\omega}{2}\left(\dfrac{3|\mathbf{h}|}{\alpha}-\left(\dfrac{|\mathbf{h}|}{\alpha}\right)^3\right), & \text{ha } |\mathbf{h}| \leqq \alpha \\ \omega, & \text{ha } |\mathbf{h}| > \alpha, \end{cases} \qquad \text{(szférikus)}$$

ahol $|\mathbf{h}|$ a $\mathbf{h}$ vektor hosszát jelöli, $\omega \geqq 0$ állandó.

Foglalkozzunk először az 1. feladattal. Keressük az y_i közelítő függvényértékek segítségével az $f_N(\mathbf{x}^*)$ becslést a lineáris

$$f_N(\mathbf{x}^*) = \lambda_1 y_1 + \ldots + \lambda_N y_N \tag{2.4}$$

alakban, ahol a λ_i együtthatók egyelőre ismeretlenek. A (2.4) becslésről feltesszük, hogy torzítatlan, és a becslés hibájának szórása minimális. A torzítatlansági feltétel azt jelenti, hogy

$$E[f_N(\mathbf{x}^*)] = E[f(\mathbf{x}^*)],$$

azaz

$$\sum_{i=1}^{N} \lambda_i E(y_i) = E[f(\mathbf{x}^*)],$$

amelyből

$$\sum_{i=1}^{N} \lambda_i \sum_{j=1}^{J} a_j \varphi_j(\mathbf{x}_i) = \sum_{j=1}^{J} a_j \varphi_j(\mathbf{x}^*).$$

Ez nyilvánvalóan teljesül akkor, ha $j=1, 2, \ldots, N$ esetén ([5])

$$\sum_{i=1}^{N} \lambda_i \varphi_j(\mathbf{x}_i) = \varphi_j(\mathbf{x}^*). \tag{2.5}$$

Tegyük fel, hogy $\varphi_1(\mathbf{x}) \equiv 1$, így $j=1$ esetén a (2.5) feltétel

$$\sum_{i=1}^{N} \lambda_i = 1$$

alakú.

Tekintsük ezután a becslés varianciáját. Egyszerű számolással látható, hogy

$$E\left[\left(f_N(\mathbf{x}^*) - \sum_{i=1}^{N} \lambda_i y_i\right)^2\right] = \operatorname{Var}\left(f_N(\mathbf{x}^*) - \sum_{i=1}^{N} \lambda_i y_i\right) =$$

$$= \operatorname{Var}\left(\sum_{i=1}^{N} \lambda_i (f_N(\mathbf{x}^*) - y_i)\right) = \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j \operatorname{Cov}\left(f_N(\mathbf{x}^*) - y_i,\ f_N(\mathbf{x}^*) - y_j\right),$$

ahol Cov a kovariancia jele. Vegyük észre, hogy

$$\operatorname{Cov}\left(f_N(\mathbf{x}^*) - y_i, f_N(\mathbf{x}^*) - y_j\right) =$$

$$= \frac{1}{2}\left[-\operatorname{Var}\left((f_N(\mathbf{x}^*) - y_i) - (f_N(\mathbf{x}^*) - y_j)\right) + \operatorname{Var}\left(f_N(\mathbf{x}^*) - y_i\right) + \operatorname{Var}\left(f_N(\mathbf{x}^*) - y_j\right)\right] =$$

$$= -\gamma(\mathbf{x}_i - \mathbf{x}_j) + \gamma(\mathbf{x}^* - \mathbf{x}_i) + \gamma(\mathbf{x}^* - \mathbf{x}_j).$$

Ekkor a becslési variancia minimalizálása a (2.5) feltételekkel a *Lagrange-féle elvvel* ([6]) lineáris egyenletrendszerre vezet:

$$\begin{aligned} \sum_{k=1}^{N} \lambda_k \gamma(\mathbf{x}_i - \mathbf{x}_k) &= \gamma(\mathbf{x}_i - \mathbf{x}^*) + \sum_{j=1}^{J} \mu_j \varphi_j(\mathbf{x}_i) \quad (1 \leqq i \leqq N), \\ \sum_{i=1}^{N} \lambda_i \varphi_j(\mathbf{x}_i) &= \varphi_j(\mathbf{x}^*) \quad (1 \leqq j \leqq J), \end{aligned} \tag{2.6}$$

ahol $\mu_j \geqq 0$ állandók.

Ezt az egyenletrendszert az *univerzális-Kriging-egyenletrendszernek* is nevezik a gyakorlatban ([3]). Ennek megoldása adja az ismeretlen λ_i együtthatíkat, valamint az optimumhelyen a becslési hiba varianciája:

$$E\left[\left(f_N(\mathbf{x}^*) - f(\mathbf{x}^*)\right)^2\right] = \sum_{i=1}^{N} \lambda_i \gamma(\mathbf{x}^* - \mathbf{x}_i) + \sum_{j=1}^{J} \mu_j \varphi_j(\mathbf{x}^*). \tag{2.7}$$

A 2. feladat esetében is hasonló a módszer alakja. Itt is (2.6) típusú egyenlet-

rendszert kell megoldanunk:

$$\sum_{k=1}^{N} \lambda_k \gamma(\mathbf{x}_i - \mathbf{x}_k) = \bar{\gamma}(\mathbf{x}_i) + \sum_{j=1}^{J} \mu_j \varphi_j(\mathbf{x}_i) \quad (1 \leqq i \leqq N)$$

$$\sum_{i=1}^{N} \lambda_i \varphi_j(\mathbf{x}_i) = \bar{\varphi}_j \quad (1 \leqq j \leqq J) \tag{2.8}$$

a becslési hiba varianciája pedig:

$$\sum_{i=1}^{N} \lambda_i \bar{\gamma}(\mathbf{x}_i) + \sum_{j=1}^{J} \mu_j \bar{\varphi}_j - \bar{\gamma}, \tag{2.9}$$

ahol

$$\bar{\gamma}(\mathbf{x}_i) = \frac{1}{|D|} \int_D \gamma(\mathbf{x}_i - \mathbf{x}^*)\, d\mathbf{x},$$

$$\bar{\varphi}_j = \frac{1}{|D|} \int_D \varphi_j(\mathbf{x})\, d\mathbf{x},$$

$$\bar{\gamma} = \frac{1}{|D|^2} \int_D \int_D \gamma(\mathbf{x} - \mathbf{x}')\, d\mathbf{x}\, d\mathbf{x}'.$$

Megjegyezzük, hogy a (2.6) és (2.9) egyenletrendszerek megoldására gyakorlatban is jól alkalmazható módszerek állnak rendelkezésünkre ([7], [8]).

3. Az 1. feladat módszerének konvergenciája

Az 1. feladat esetére bemutatott módszerre tegyük fel a következőket:

(i) az y_i megfigyelések pontosak, azaz $n(\mathbf{x}_i)=0$ $(1 \leqq i \leqq N)$;
(ii) $\gamma(\mathbf{0})=0$ és γ folytonos az origó környezetében;
(iii) $J=1$, $\varphi_1(\mathbf{x}) \equiv 1$;
(iv) a $\gamma(\mathbf{h})$ variogram ismert és pontos.

3.1. TÉTEL. Az f függvény értelmezési tartományát D-vel jelölve tegyük fel, hogy a végtelen $\mathbf{x}_1, \mathbf{x}_2, \ldots$ sorozat D-ben sűrű. Ekkor tetszőleges $\mathbf{x}^* \in D$ esetén

$$E\left[\left(f(\mathbf{x}^*) - f_N(\mathbf{x}^*)\right)^2\right] \to 0, \quad \text{ha} \quad N \to \infty.$$

Bizonyítás. A feltételek alapján tetszőleges $1 \leqq i \leqq N$ mellett a $\lambda_1=0, \ldots, \lambda_{i-1}=0, \lambda_i=1, \lambda_{i+1}=0, \ldots, \lambda_N=0$ együtthatók kielégítik a (2.5) feltételt, így

$$E\left[\left(f(\mathbf{x}^*) - f_N(\mathbf{x}^*)\right)^2\right] \leqq E\left[\left(f(\mathbf{x}^*) - f(\mathbf{x}_i)\right)^2\right] = 2\gamma(\mathbf{x}^* - \mathbf{x}_i). \tag{3.1}$$

Jelölje most $\mathbf{x}^*(N)$ az $\mathbf{x}^*$-hoz legközelebbi pontot az $\mathbf{x}_1, \ldots, \mathbf{x}_N$ alappontok közül, ekkor nyilvánvalóan $N \to \infty$ esetén $\mathbf{x}^*(N) \to \mathbf{x}^*$, így (3.1) alapján

$$E\left[\left(f(\mathbf{x}^*) - f_N(\mathbf{x}^*)\right)^2\right] \leqq 2\gamma\left(\mathbf{x}^* - \mathbf{x}^*(N)\right) \to 0.$$

KÖVETKEZMÉNY. Vezessük be a következő jelölést:

$$\varepsilon_N = \max_{\mathbf{x} \in D} \min_{1 \leq i \leq N} |\mathbf{x} - \mathbf{x}_i|,$$

ekkor (3.1) alapján nyilvánvalóan

$$E[(f(\mathbf{x}^*) - f_N(\mathbf{x}^*))^2] \leqq 2\gamma(\varepsilon_N),$$

ahol most feltesszük, hogy γ csak a $|\mathbf{h}|$ vektorhossztól függ. Ha például D kétdimenziós, és olyan négyzetrács pontjaiban ismerjük a függvényértékeket, amely egybevágó, A oldalú négyzetekből áll, akkor nyilvánvalóan

$$\varepsilon_N = \frac{A}{\sqrt{2}}.$$

Ha D háromdimenziós, és A oldalú kockarács csúcspontjai az $\mathbf{x}_i$ alappontok, akkor ez esetben

$$\varepsilon_N = \frac{A\sqrt{3}}{2}.$$

Tekintsük ezután az általános esetet, amikor $J \geqq 1$. Jelölje $\mathbf{x}_i^*(N)$ az $\mathbf{x}^*$-hoz az i-edik legközelebbi alappontot az $\mathbf{x}_1, \ldots, \mathbf{x}_N$ pontok közül. Tegyük fel ismét, hogy az $\mathbf{x}_1, \mathbf{x}_2, \ldots$ végtelen sorozat sűrű D-ben, valamint a

$$\boldsymbol{\Phi}_N = (\varphi_j(\mathbf{x}_i(N)))_{j,i=1}^J$$

mátrix invertálható. Legyen továbbá $\boldsymbol{\lambda}_N$ a

$$\boldsymbol{\Phi}_N \boldsymbol{\lambda}_N = \boldsymbol{\pi}$$

egyenletrendszer megoldása, ahol $\boldsymbol{\pi} = (\varphi_1(\mathbf{x}^*), \ldots, \varphi_J(\mathbf{x}^*))^T$. Ekkor $\boldsymbol{\lambda}_N$ komponensei nyilvánvalóan kielégítik a (2.5) torzítatlansági feltételeket, valamint ekkor a

$$\boldsymbol{\lambda}_N = (\lambda_1^N, \ldots, \lambda_J^N)$$

jelöléssel

$$E[(f(\mathbf{x}^*) - f_N(\mathbf{x}^*))^2] \leqq E\left[\left(f(\mathbf{x}^*) - \sum_{j=1}^J \lambda_j^N f(\mathbf{x}_j(N))\right)^2\right] =$$

(3.2)

$$= E\left[\left(\sum_{j=1}^J \lambda_j^N (f(\mathbf{x}^*) - f(\mathbf{x}_j(N)))\right)^2\right] \leqq J \cdot \sum_{j=1}^J |\lambda_j^N|^2 \gamma(\mathbf{x}^* - \mathbf{x}_j(N)).$$

Itt a legutóbbi összefüggésben felhasználtuk, hogy tetszőleges $\alpha_1, \ldots, \alpha_J$ valószínűségi változók esetén

$$E\left[\left(\sum_{j=1}^J \alpha_j\right)^2\right] \leqq J \cdot \sum_{j=1}^J E(\alpha_j^2), \tag{3.3}$$

amely a számtani és a négyzetes közép közvetlen összehasonlításából azonnal leolvasható.

A (3.2) összefüggés felhasználásával bebizonyítjuk a következő tételt.

3.2. TÉTEL. Tegyük fel, hogy a 3.1. tétel (i), (ii), (iv) feltételei teljesülnek, a Φ_N mátrix invertálható, az $\mathbf{x}_1, \mathbf{x}_2, \ldots$ végtelen pontsorozat sűrű D-ben, valamint a λ_N vektorok hossza N-től független korlát alatt marad. Ekkor $N \to \infty$ esetén

$$E[(f(\mathbf{x}^*) - f_N(\mathbf{x}^*))^2] \to 0.$$

Bizonyítás. Elegendő azt belátnunk, hogy a (3.2) jobb oldala $N \to \infty$ esetén zérushoz tart. Minthogy a feltételek alapján $J, |\lambda_j^N|$ korlátos, azt kell belátnunk, hogy

$$\gamma(\mathbf{x}^* - \mathbf{x}_j(N)) \to 0,$$

azaz $\mathbf{x}_j(N) \to \mathbf{x}^*$. Minthogy itt $j \leqq J$, ezért az $\mathbf{x}_1, \mathbf{x}_2, \ldots$ pontsorozat D-beli sűrűségéből közvetlenül leolvasható az állítás.

A következőkben az (i) feltételtől tekintsünk el, és vizsgáljuk meg az $n(\mathbf{x}_i)$ zajok, hibák hatását is. A pontos függvényértékekből adódó becslés:

$$f_N(\mathbf{x}^*) = \sum_{i=1}^{N} \lambda_i f(\mathbf{x}_i),$$

ahol λ_i a (2.6) egyenletrendszer megoldása. Ugyanígy a pontatlan függvényértékekből adódó becslés alakja

$$\tilde{f}_N(\mathbf{x}^*) = \sum_{i=1}^{N} \lambda_i [f(\mathbf{x}_i) + n(\mathbf{x}_i)],$$

ahol ugyanazok a λ_i együtthatók szerepelnek, mint $f_N(\mathbf{x}^*)$ becslésében. Ezen utóbbi állítás azonnal leolvasható abból a tényből, hogy pontosan ismert $\gamma(\mathbf{h})$ variogramfüggvény esetén a (2.6) egyenletrendszer nem függ a függvényértékektől. Ekkor pedig a hiba:

$$\varepsilon = \tilde{f}_N(\mathbf{x}^*) - f_N(\mathbf{x}^*) = \sum_{i=1}^{N} \lambda_i n(\mathbf{x}_i).$$

Ennek várható értéke:

$$E(\varepsilon) = \sum_{i=1}^{N} \lambda_i E[n(\mathbf{x}_i)] = 0,$$

szórásnégyzetére pedig fennáll, hogy

$$E(\varepsilon^2) = E\left[\left(\sum_{i=1}^{N} \lambda_i n(\mathbf{x}_i)\right)^2\right] = \sum_{i=1}^{N} |\lambda_i|^2 \sigma^2. \tag{3.4}$$

4. A 2. feladat módszerének vizsgálata

Mielőtt a 2. feladat módszerét megvizsgálnánk, egy fontos észrevételt kell tennünk. Jelöljük az 1. feladat megoldásában szereplő λ_i együtthatókat a $\lambda_i(\mathbf{x}^*)$ szimbólummal, hogy az együtthatóknak az $\mathbf{x}^*$-tól való függését illusztráljuk. Ha λ_i jelöli a 2. feladat megoldásában szereplő együtthatókat, akkor $i = 1, 2, \ldots, N$

esetén

$$\lambda_i = \frac{1}{|D|}\int_D \lambda_i(\mathbf{x})\,d\mathbf{x}. \tag{4.1}$$

Ez az észrevétel közvetlenül leolvasható abból, hogy a λ_i együtthatókat szolgáltató egyenletrendszer együttható-mátrixa azonos a két feladat esetében, és a 2. feladat esetén a jobb oldal elemeinek átlagértékei szerepelnek, amely olyan lineáris funkcionált jelent, amely konstanst változatlanul hagy. Jelölje ezután f_N a 2. feladatban nyert becslést. Ekkor a $\tilde{t}$ szimbólummal jelölve az átlagképzés (4.1)-ben adott funkcionálját:

$$E[(f_N - \tilde{t}\,(f(\mathbf{x})))^2] = E\left[\left(\tilde{t}\left(\sum_{i=1}^{N}\lambda_i(\mathbf{x})\,y_i - f(\mathbf{x})\right)\right)^2\right] \leqq \tag{4.2}$$

$$\leqq \|\tilde{t}\,\|^2 \sup_{\mathbf{x}\in D} E[(f_N(\mathbf{x}) - f(\mathbf{x}))^2]\cdot|D|,$$

ahol $\|\tilde{t}\|$ a $\tilde{t}$ normáját jelöli, amelyet például a *Cauchy—Schwarz-féle egyenlőtlenséggel* becsülhetünk majd meg. Nyilvánvalóan

$$\|\tilde{t}\,\| = \sup\frac{|\tilde{t}\,(f(\mathbf{x}))|}{\|f(\mathbf{x})\|_{L_2}},$$

ahol

$$\|f(\mathbf{x})\|_{L_2} = \left\{\int_D |f(\mathbf{x})|^2\,d\mathbf{x}\right\}^{\frac{1}{2}}.$$

Ekkor pedig

$$\|\mathscr{F}\| = \sup\frac{\frac{1}{|D|}\int_D 1\cdot f(\mathbf{x})\,d\mathbf{x}}{\|f(\mathbf{x})\|_{L_2}} \leqq \sup\frac{\frac{1}{|D|}\left\{\int_D 1\,d\mathbf{x}\right\}^{\frac{1}{2}}\cdot\left\{\int_D |f(\mathbf{x})|^2\,d\mathbf{x}\right\}^{\frac{1}{2}}}{\|f(\mathbf{x})\|_{L_2}} = \frac{1}{\sqrt{|D|}},$$

amelynek felhasználásával a (4.2) egyenlőtlenségek alapján

$$E[(f_N - \tilde{t}\,(f(\mathbf{x})))^2] \leqq \sup_{\mathbf{x}\in D} E[(f_N(\mathbf{x}) - f(\mathbf{x}))^2], \tag{4.3}$$

amellyel az átlagérték becslési hibájának vizsgálatát visszavezettük a pontbecslések esetére; így az előző paragrafus eredményeit minden további nélkül alkalmazni tudjuk.

IRODALOM

[1] DELHOMME, J. P., "Kriging in hydrosciences", *Advances in Water Resources* **1** (1978) 251–266.
[2] GAMBOLETI, G. and VOLPI, G., "Groundwater contour mapping in Venice by stochastic interpolators", *Water Resources Research* **15** (1979) 281–290.
[3] JOURNEL, A. G. and HUIJBREGTS, CH. J., *Mining Geostatistics* (Academic Press, London, 1978).
[4] MATHERON, G., "The intrinsic random functions and their applications", *Advances in Applied Probability* **5** (1973) 439–468.
[5] RÉNYI, A., *Valószínűségszámítás* (Tankönyvkiadó, Budapest, 1966).

[6] SZÉP, J., *Analízis* (Közgazdasági és Jogi Könyvkiadó, Budapest, 1974).
[7] SZIDAROVSZKY, F., *Bevezetés a numerikus módszerekbe* (Közgazdasági és Jogi Könyvkiadó, Budapest, 1974).
[8] SZIDAROVSZKY, F. and YAKOWITZ, S., *Principles and Procedures in Numerical Analysis* (Plenum Press, London, 1978).

(Beérkezett: 1981. augusztus 28.)

MOLNÁR SÁNDOR
KÖZPONTI BÁNYÁSZATI FEJLESZTÉSI INTÉZET
1027 BUDAPEST, VARSÁNYI I. U. 40—44.

SZIDAROVSZKY FERENC
KERTÉSZETI EGYETEM SZÁMÍTÓKÖZPONT
1118 BUDAPEST, MÉNESI ÚT 44.

SOME PROPERTIES OF THE KRIGING METHOD

S. MOLNÁR S. YAKOWITZ F. SZIDAROVSZKY

In our paper we investigate some mathematical properties of the general and classical Kirging method. When we increase the number of base points to infinity, we can proove the convergence of the methods. We investigate the effect of non exact function values, too.

Alkalmazott Matematikai Lapok 7 (1981) 279—285

KONVEX, DIFFERENCIÁLHATÓ FÜGGVÉNY FOKOZATOS KÖZELÍTÉSE OPTIMÁLIS MÉRÉSI HELYEK MEGHATÁROZÁSA ALAPJÁN

TÖRÖK TAMÁS

Esztergom

A dolgozatban konvex, differenciálható függvény fokozatos közelítését adjuk az optimális mérési hely matematikai definiálása és meghatározása alapján. Ezen eljárást általánosítjuk zárt görbék egy osztályára, majd speciális térbeli tartományokra.

1. A feladat matematikai körvonalazása

Legyen az $f(x)$ differenciálható függvény az $[a, b]$ zárt intervallumon konvex. Tételezzük fel, hogy $f(x)$, illetve deriváltjának, $f'(x)$-nek értékét az intervallum két végpontjában ismerjük:

$$y_a = f(a), \quad y_b = f(b), \quad m_a = f'(a), \quad m_b = f'(b).$$

Tegyük fel továbbá, hogy $f(x)$ és $f'(x)$ értéke az intervallum bármely pontjában megismerhető úgy, hogy erre mérést végzünk.

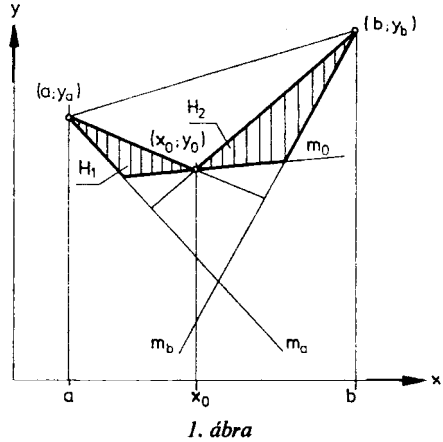
Célunk ezen mérési hely optimális megválasztása olyan értelemben, hogy $f(x)$ alakjára vonatkozóan a legtöbb információt szerezzük. Az optimális mérési helyet egzakt módon a későbbiekben fogjuk definiálni.

További feladat a fentiek általánosítása: n számú mérés alapján az $(n+1)$-edik mérési hely optimális megválasztása.

2. Az optimális mérési hely fogalma és meghatározása

Rögzítsünk az $[a, b]$-n belül egy x_0 mérési helyet. Tételezzünk fel x_0-ban $y_0 = f(x_0)$ függvényértéket és $m_0 = f'(x_0)$ deriváltértéket.

Ekkor a konvexitás miatt $f(x)$ csak a $H_1 \cup H_2$ tartományban helyezkedhet el (1. ábra).



1. ábra

Jelöljük a H_1 háromszög területét T_1-gyel, a H_2 háromszög területét pedig T_2-vel.

Az x_0 mérési helyhez olyan függvény- és deriváltértéket fogunk hozzárendelni, hogy $f(x)$ a lehető legnagyobb területű tartományban helyezkedhessen el, legkevésbé korlátozva ezzel az y_a, y_0 és y_b-n kívüli függvényértékeket.

Tehát egy rögzített alapponthoz (x_0) tartozó feltételezett függvény-, ill. deriváltérték azzal a követelménnyel lesz egyértelműen meghatározva, hogy:

$$T(x_0, y, m) = T_1(x_0, y, m) + T_2(x_0, y, m)$$

legyen maximális y-ban és m-ben. Vezessük be a $T(x_0) = \max_{y,m} T(x_0, y, m)$ jelölést. Ezekután definiálhatjuk az optimális mérési helyet: Az x_{opt} optimális mérési hely $[a, b]$-ben, ha

$$T(x_{opt}) < T(x), \quad x \in [a, b]. \tag{2.1}$$

Vagyis x_{opt} a $T(x)$ függvény szigorú értelemben vett minimumhelye az $[a, b]$-n. A későbbiekben látni fogjuk, hogy a fent definiált x_{opt} egyértelműen meghatározott, tehát a „$<$" reláció jogos.

Most pedig térjünk rá az optimális mérési hely meghatározására.

Az egyszerűség kedvéért legyen:

$$(a, y_a) = (0, 0),$$
$$(b, y_b) = (x_2, y_2)$$

és jelöljük m_a-t m_1-gyel, m_b-t pedig m_2-vel. Ez egy egyszerű eltolási transzformáció, amit majd a számítások végén figyelembe kell vennünk.

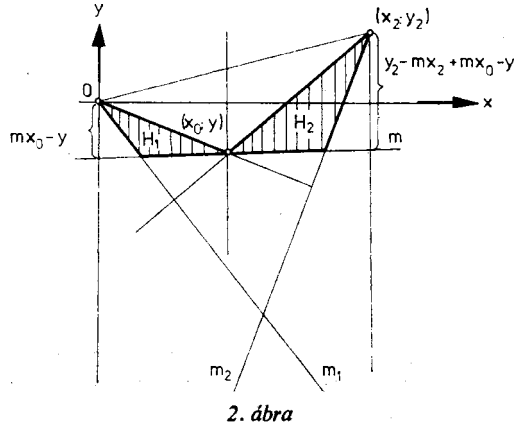
A 2. ábra segítségével (m_1, m_2, m a megfelelő egyenesek iránytangenseit jelöli) T_1, ill. T_2-re a következő kifejezéseket kapjuk:

$$T_1(x_0, y, m) = \frac{(mx_0 - y)(m_1 x_0 - y)}{2(m_1 - m)},$$

$$T_2(x_0, y, m) = \frac{(y_2 - mx_2 + mx_0 - y)(y - m_2 x_0 + m_2 x_2 - y_2)}{2(m_2 - m)}.$$

$$\frac{\partial T}{\partial y} = 0 \Rightarrow \frac{2y - x_0(m_1 + m)}{m_1 - m} = \frac{2y - 2y_2 + (x_2 - x_0)(m_2 + m)}{m_2 - m} \tag{I}$$

$$\frac{\partial T}{\partial m} = 0 \Rightarrow \frac{m_1 x_0 - y}{m_1 - m} = \frac{y - y_2 + m_2(x_2 - x_0)}{m_2 - m}. \tag{II}$$



2. ábra

Ez utóbbi kétismeretlenes egyenletrendszer (I) és (II) (m, y) megoldásának a $T(x_0, y, m)$ függvénybe való helyettesítésével kapjuk:

$$T(x_0) = \frac{1}{32}[x_0^2(16m_2 - 16m_1) + x_0(16y_2 - 24m_2x_2 + 8m_1x_2) + 9m_2x_2^2 - m_1x_2^2 - 8y_2x_2].$$

Ez a függvény pedig $m_2 > m_1$ miatt egy felfelé nyíló parabola, tehát minimuma van.

Számítsuk ki a minimum helyét és akkor megkapjuk x_{opt} értékét.

$$\frac{\partial T}{\partial x_0} = 0 \Rightarrow \underset{[0,x_2]}{x_{opt}} = \frac{x_2(3m_2 - m_1) - 2y_2}{4(m_2 - m_1)}. \tag{2.2}$$

Az $[a, b]$-re vonatkozó optimumhely a levezetés elején alkalmazott eltolás miatt:

$$\underset{[a,b]}{x_{opt}} = \frac{(b-a)(3m_b - m_a) - 2(y_b - y_a)}{4(m_b - m_a)} + a. \tag{2.3}$$

3. Az optimális mérési hely tulajdonságai

Vizsgáljuk meg x_{opt} tulajdonságait az egyszerűség kedvéért a $[0, x_2]$ intervallumon.

1. Nem tudunk optimális mérési helyet keresni, ha mindkét határoló egyenes függőleges $(m_1 = m_2)$, hiszen ebben az esetben a függvény egy végtelen hosszú sávban helyezkedhetne el.

2. $x_{opt} \in (0, x_2)$ (ez nyilván jogos elvárás is).

Bizonyítás:

a) Az $\frac{x_2(3m_2 - m_1) - 2y_2}{4(m_2 - m_1)} > 0$ egyenlőtlenség az $x_2 > 0$ és $m_1 < \frac{y_2}{x_2} < m_2$ relációkból közvetlenül adódik.

b) Az $\frac{x_2(3m_2 - m_1) - 2y_2}{4(m_2 - m_1)} < x_2$ egyenlőtlenség a)-hoz hasonlóan látható be.

3. $\lim_{m_1 \to -\infty} x_{opt} = \frac{x_2}{4}$.

Vagyis, ha a bal oldali egyenes függőleges $(m_1 \to -\infty)$, akkor x_{opt} az intervallum bal végpontjához közelebb eső negyedelőpontja.

4. $\lim_{m_2 \to \infty} x_{opt} = \frac{3}{4}x_2$.

5. Ha $m_1 = -m_2$ és $y_2 = 0$, akkor $x_{opt} = \frac{x_2}{2}$.

Ez a tulajdonság azt jelenti, hogy ha a befoglaló háromszög egyenlő szárú, és szimmetriatengelye függőleges, akkor az intervallum felénél kell mérni.

6. Az optimális mérési helyen figyelembe vett érintő párhuzamos az adott két pontot összekötő egyenessel:

$$m(x_{opt}) = \frac{y_2}{x_2}.$$

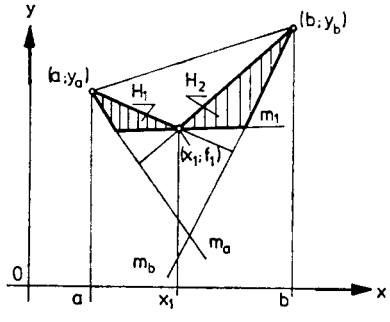
7. $T_1(x_{opt}, y(x_{opt}), m(x_{opt})) = T_2(x_{opt}, y(x_{opt}), m(x_{opt}))$.

Egyszerű számolással ellenőrizhető.

4. Az $(n+1)$-edik optimális mérési hely meghatározása az első n figyelembevételével

Tételezzük fel, hogy az első optimális mérési hely, x_1 már meghatározott.

Végezzük el a mérést az x_1 pontban. Tapasztaljunk itt f_1 függvényértéket és m_1 deriváltértéket. Akkor a 3. ábráról leolvasható, hogy $f(x)$ konvexitása miatt csak a $H_1 \cup H_2$ tartományban helyezkedhet el.



3. ábra

Jellemezzük az x_1 optimális mérési helyet a $H_1 \cup H_2$ tartomány mértékével. Nyilván ez a T_1 terület összefüggésben van a függvény megismerésével, hiszen ha ez a terület kicsi, akkor $f(x)$ csak egy szűk tartományban helyezkedhet el $[a,b]$-n. Határozzuk most meg a második optimális mérési helyet, x_2-t.

Legyen $I_1=(a, x_1)$, $I_2=(x_1, b)$.

Jelöljük $x_2^{(I_1)}$-gyel I_1-en, $x_2^{(I_2)}$-vel pedig I_2-n a (2.1) alapján meghatározott optimális mérési helyet. Tegyük az alábbi hozzárendeléseket:

$$x_2^{(I_1)} \to T^{(I_1)}, \quad x_2^{(I_2)} \to T^{(I_2)}.$$

Definiáljuk x_2-t a részoptimumok közül azzal, amelyhez tartozó terület nagyobb.

Jussunk el ily módon az első n $(x_1, x_2, \ldots, x_n)$ optimális mérési hely meghatározásáig, és rendezzük ezeket növekvő sorrendbe:

$$\begin{array}{ccccccccc} & & T_1' & & & & T_n' & & \\ & & \uparrow & & & & \uparrow & & \\ a & < & x_1' & < & \ldots & < & x_n' & < & b \\ \downarrow & & \downarrow & & & & \downarrow & & \downarrow \\ m_a & & m_1' & & & & m_n' & & m_b \end{array}$$

$$I_1 = (a, x_1'),\ I_2 = (x_1', x_2'),\ \ldots,\ I_{n+1} = (x_n', b).$$

Jelölje az egyes I_k $(k=1, 2, \ldots, n+1)$ részintervallumokba eső optimális mérési helyeket:

$$x_{n+1}^{(I_1)} \in I_1,\ x_{n+1}^{(I_2)} \in I_2,\ \ldots,\ x_{n+1}^{(I_{n+1})} \in I_{n+1}$$

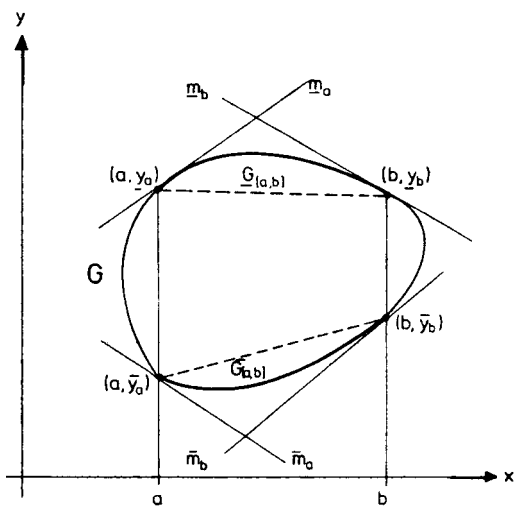
$$x_{n+1}^{(I_k)} \to T^{(I_k)} \quad (k = 1, 2, \ldots, n+1).$$

Ha $T^{(I_p)} = \max_{1 \le j \le n+1} T^{(I_j)}$, akkor $x_{n+1}^{(I_p)}$ legyen az $(n+1)$-edik optimális mérési hely az $[a, b]$-ben.

Ha valamely mérési helyen $f'(x)$ értéke nem ismerhető meg, akkor a rákövetkező optimális mérési hely meghatározására ismertetett eljárás szintén alkalmazható azzal a különbséggel, hogy a függvény nagyobb elhelyezkedési tartományát kell figyelembe vennünk (ui. az érintő egyenes nem vág le a befoglaló háromszögekből).

5. Optimális mérési hely zárt görbék egy osztályára

Tételezzük fel, hogy a G zárt görbe felbontható egy $\bar{G}$ konvex, és egy $\underline{G}$ konkáv ívre (4. ábra).



4. ábra

Ismerjük a G függvény, és deriváltjának értékeit az a, ill. b mérési helyeken (ez 4-4 adat, mivel a függvény kétértékű). Tartozzon az $[a, b]$ intervallumhoz a $\bar{G}_{[a,b]}$, ill. a $\underline{G}_{[a,b]}$ ívdarab. A 2. fejezetben tárgyaltak alapján mindkét ívdarabhoz tartozik egy-egy optimális mérési hely. Az $[a, b]$-re vonatkozó optimális mérési hely definiálásánál most az alsó (alulról konvex), és a felső (felülről konvex) ív egy x_0 mérési helyhez tartozó függvényértékeinek bizonytalanságát egyidejűleg kell figyelembe venni (hiszen előfordulhat, hogy csak a felső ívhez tartozó optimális mérési hely az alsó íven közel nulla megismerést eredményez).

Ebben az esetben egy rögzített x_0 alapponthoz tartozó feltételezett függvény-, ill. deriváltértékek a $\bar{G}_{[a,b]}$, ill. a $\underline{G}_{[a,b]}$ íven azzal a követelménnyel lesznek egyértelműen meghatározva, hogy:

$$T(x_0, \bar{y}, \bar{m}, \underline{y}, \underline{m}) = \bar{T}(x_0, \bar{y}, \bar{m}) + \underline{T}(x_0, \underline{y}, \underline{m})$$

legyen maximális $\underline{y}, \underline{m}, \bar{y}, \bar{m}$-ben.

($\overline{T}(x_0, \bar{y}, \overline{m})$ és $\underline{T}(x_0, \underline{y}, \underline{m})$ hasonlóan értelmezendők, mint a 2. fejezetben a $T(x_0, y, m)$ függvény.)

Vezessük be a $T(x_0) = \max_{\bar{y}, \overline{m}, \underline{y}, \underline{m}} T(x_0, \bar{y}, \overline{m}, \underline{y}, \underline{m})$ jelölést. Ezek után nevezzük x_{opt}-t optimális mérési helynek $[a, b]$-n, ha:

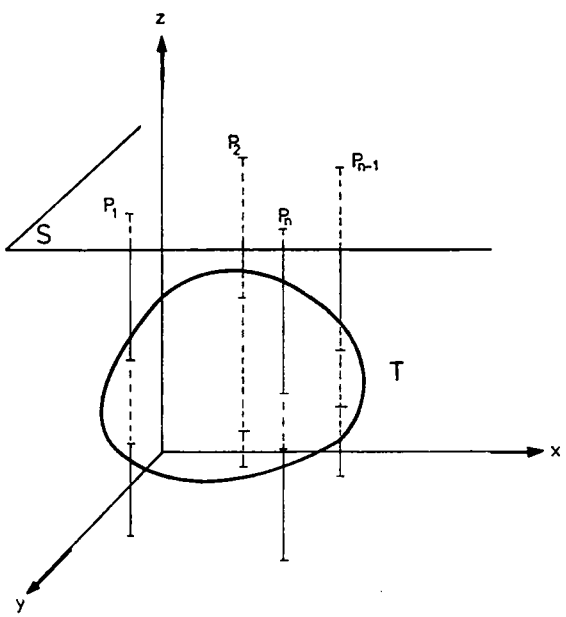
$$T(x_{opt}) \prec T(x), \quad x \in [a, b]. \tag{5.1}$$

A 4. ábra jelöléseit figyelembe véve, a részletszámítások mellőzésével kapjuk:

$$x_{\substack{opt \\ [a,b]}} = \frac{(b-a)[3(\overline{m}_b - \underline{m}_b) - (\overline{m}_a - \underline{m}_a)] - 2[(\bar{y}_b - \bar{y}_a) - (\underline{y}_b - \underline{y}_a)]}{4[(\overline{m}_b - \underline{m}_b - (\overline{m}_a - \underline{m}_a)]} + a. \tag{5.2}$$

6. A módszer térbeli alkalmazásának lehetőségéről

Helyezzük el a T geometriai alakzatot az xyz koordinátarendszerbe (5. ábra).



5. ábra

Tételezzük fel, hogy T-nek az xy síkra merőleges síkmetszetei teljesítik az 5. fejezet kiinduló feltételeit, vagyis egy tetszőleges ilyen metszet felbontható egy konvex és egy konkáv ívre.

Az xy síkkal párhuzamos S síkon vegyünk fel n számú, $(P_1, P_2, \ldots, P_n)$ mérési helyet, és ezen pontokon keresztül húzzunk merőlegeseket az xy síkra.

Ha a döféspontok koordinátái és a hozzájuk tartozó, T-re vonatkozó érintősíkok megismerhetők, akkor válasszunk ki két tetszőleges mérési helyet S-en, pl.

P_i-t és P_j-t. Jelöljük a hozzájuk tartozó, xy síkra merőleges egyenesekre fektetett síkot S_{ij}-vel. Ha ezt a metszetet kinagyítjuk, akkor a 4. ábrához hasonlót kapunk. A döféspontokhoz tartozó érintősíkok S_{ij}-vel alkotott metszésvonalai szolgáltatják majd az érintő egyeneseket az S_{ij} síkban. Így, az 5. fejezetben tárgyaltak alapján a P_iP_j-n egy optimális mérési helyhez juthatunk.

Ha ezt valamennyi (P_i, P_j) pontpárra elvégezzük, akkor $\binom{n}{2}$ számú, egy-egy intervallumhoz tartozó optimális mérési helyet kapunk, amelyek sorrendbe állíthatók a 4. fejezetben alkalmazott rangsoroláshoz hasonlóan.

Tehát a $(P_1, P_2, ..., P_n)$ pontok konvex burkolója által bezárt területen egy újabb (számított) mérési helyet határozhatunk meg.

A dolgozat keletkezésének körülményeiről

FEUER GÁBOR 1978-ban a jelen dolgozat 1. fejezetében vázolt feladat egy megoldását adta [1]. Eljárásában az optimális mérési helyet a maximális függvényérték eltérések minimumával definiálta.

Eredményül az optimális mérési helynek egy implicit, és rendkívül bonyolult kifejezését kapta, amelyből a mérési helynek bizonyos elemi tulajdonságai sem állapíthatók meg. Dolgozatában — valószínűleg az előbbiek miatt — nem tesz említést eljárásának zárt görbékre és térbeli tartományokra vonatkozó alkalmazhatóságáról.

A problémával úgy kerültem kapcsolatba, hogy az általa definiált optimális mérési hely számítógépes meghatározásával kezdtem el foglalkozni. Ennek eredménytelensége inspirált a probléma újbóli átgondolására, majd a jelen cikk megírására.

IRODALOM

[1] FEUER, G.: „Adaptív algoritmus ásványvagyonbecslésnél", KBFI kézirat.

(Beérkezett: 1980. február 11.)

TÖRÖK TAMÁS
NIM TOVÁBBKÉPZŐ KÖZPONT
2511 ESZTERGOM-KERTVÁROS, WESSELÉNYI U. 35—39.

SUCCESSIVE APPROXIMATION OF CONVEX DIFFERENCIABLE FUNCTION BY DETERMINATION OF OPTIMAL MEASURING PLACE

T. TÖRÖK

The present study deals in the first place with solving the original and theoretical problem mentioned in the tittle. We can see the planar and spatial application of this method on case of realization of certain special conditions.

Alkalmazott Matematikai Lapok 7 (1981) 287—309

SZÁMLÁLÁSOS CIKLUSOKBÓL FELÉPÍTHETŐ PROGRAMOK MAGNYELVEINEK SZERKEZETÉRŐL

DÖRNYEI ÁGNES

Budapest

PRATT [9] értelmezi tetszőleges (gráf alakban megadott) program „vezérlőrészre" és „magrészre" való felbontását, valamint e felbontás alapján a program „magnyelvét", mint a program szimbolikus kiszámítási (végrehajtási) sorozatai közül a végeseknek a halmazát. A magnyelveknek a programozás szempontjából való érdekességét az adja, hogy jól tükrözik a programok viselkedését, és ezért összefüggnek a programhelyesség-bizonyítás, a szimbolikus végrehajtással történő programtesztelés és a program-optimalizálás kérdésköreivel. A jelen dolgozat célja a magnyelvek (eddig még nem tanulmányozott) szerkezetének vizsgálata. Bebizonyítjuk, hogy az egyszerű számlálásos ciklusos URM-programok magnyelveinek osztálya valódi része a környezetfüggő nyelvek osztályának, és tartalmazás szempontjából összehasonlíthatatlan a szigorúan környezetfüggő-, a szigorúan környezetfüggetlen- és a reguláris nyelvek osztályaival.

1. Bevezetés

T. PRATT egy 1978-as cikke [9] a számítógépi programok „magrészre" és „vezérlőrészre" történő felbontásával, és e felbontásnak a különböző programtulajdonságok bizonyításában való alkalmazási lehetőségeivel foglalkozik. Dolgozatunk ebből a cikkből indul ki, ezért néhány, jelölésbeli megállapodás után, először röviden ismertetjük az ott definiált főbb fogalmakat.

A „:=" jelentése „$\overset{\text{def}}{=}$", ill. programokban értékadás lesz; „$\subseteq$" és „$\subset$" a „része", ill. „valódi része" jelölésére fog szolgálni; $N := \{0, 1, 2, 3, \ldots\}$ $N_+ := \{1, 2, 3, \ldots\}$. Valamely X halmazra, $X^* :=$ *az X fölötti véges sorozatok (szavak) halmaza,* $\lambda :=$ *az üres sorozat (üres szó),* $X^+ := X^* \setminus \{\lambda\}$ ($\lambda \in X^*$, $\varnothing^* = \{\lambda\}$, $\varnothing^+ = \varnothing$); $v, w \in X^*$ esetén v és w *összefüzését (konkatenációját)* vw jelöli; $Y, Z \subseteq X^*$-ra $YZ := \{yz \mid y \in Y, z \in Z\}$, $Y = \{y\}$ esetén $y^* := \{y\}^*$, $y^+ := \{y\}^+$, $yZ := \{y\}Z$, $Zy := Z\{y\}$; $|w| :=$ *a w sorozat (szó) hossza* ($|\lambda| = 0$), $|w|_x :=$ *az x szimbólum w-beli előfordulásainak száma.* $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2$ és $\mathcal{L}_3$ rendre a *rekurzívan felsorolható-*, a *környezetfüggő-*, a *környezetfüggetlen-*, ill. a *reguláris nyelvek (Chomsky-féle) osztályai,* e négy nyelvosztály együtt alkotja a *Chomsky-féle (nyelv-)hierarchiát* (l. pl. [6, 10, 14]).

Tekintsünk egy tetszőleges P programgráfot (azaz, gráf — blokkvázlat — alakban megadott programot). Ezt úgy bontjuk fel *magrészre* és *vezérlőrészre* (e két rész nem föltétlenül diszjunkt), hogy

— a magrészbe azon programelemek tartozzanak, amelyek a program eredményének kiszámítását végzik,

— a vezérlőrészbe pedig azok, amelyek azt határozzák meg, hogy a magutasítások milyen sorrendben hajtódjanak végre.

Nyilvánvaló, hogy P bemenő-, kimenő-, elágazó- és gyűjtőpontjai a vezérlőrészhez tartoznak, az értékadások pedig (P-beli szerepüktől függően) az egyik vagy

a másik részhez, vagy mindkettőhöz tartozhatnak. A változókat is szétválaszthatjuk *magváltozókra* és *vezérlővváltozókra* (szintén nem föltétlenül diszjunkt módon), aszerint, hogy milyen utasításokban szerepelnek.

A P programgráf ún. *magsémáját*, M_P-t, a következő módon szerkesztjük meg. Helyettesítsük a P bemenő-, kimenő-, elágazó- és gyűjtőpontjai (ezek lesznek M_P pontjai) közötti értékátadás-sorozatokat egy-egy éllel (ezek lesznek M_P élei). Címkézzünk minden ilyen élt egy (C, K_i), ill. (C, λ) alakú rendezett párral, ahol C az illető élhez tartozó vezérlő értékadások sorozata, a K_i $(i \in N_+)$ ún. *magszimbólum* pedig az ugyanezen élhez tartozó magértékadások sorozatát jelöli, ha ez a sorozat nem üres, ha pedig üres, akkor K_i helyett λ áll. (Természetesen, azonos nemüres sorozatot azonos-, különböző nem-üres sorozatot különböző magszimbólummal jelölünk.) Az így kapott M_P magséma a magszimbólumok indexei erejéig egyértelműen meg van határozva.

Az M_P magsémához hozzárendeljük a C_P *vezérlési leképezést*, amely az M_P vezérlő bemenő vektorainak (a vezérlő bemenő változók lehetséges értékrendszereinek) halmazát leképezi a K^* halmazba, ahol $K=\{K_1, K_2, \ldots, K_n\}$, és K_n az M_P-ben előforduló legnagyobb indexű magszimbólum. C_P-t így definiáljuk: ha I az M_P valamely olyan vezérlő bemenő vektora, amelynek hatására az M_P-ben létrejövő kiszámítás véges, akkor vesszük az ezen kiszámítás által M_P-ben bejárt (véges) élsorozatot, ebben minden élt helyettesítünk a hozzátartozó magszimbólummal, ill. töröljük azokat az éleket, amelyekhez λ tartozik, s az így adódó magszimbólumsorozat lesz $C_P(I)$; ha pedig I olyan vezérlő bemenő vektor, amelynek hatására végtelen kiszámítás jön létre, akkor $C_P(I)=\lambda$. A C_P értékkészletét (amely tehát K fölötti nyelv) az M_P (és egyúttal P) *magnyelvének* nevezzük (amely M_P-hez hasonlóan, a magszimbólumok indexei erejéig egyértelműen meg van határozva P által). A P program magnyelvét (a magszimbólumok valamilyen, adottnak tekintett indexezése mellett) $L(P)$-vel jelöljük.

PRATT [9] cikke egyrészt matematikai megalapozását adja a korábbi hasonló, de nem egzakt módon megfogalmazott program-felbontásokkal kapcsolatos vizsgálatoknak, másrészt a benne tárgyalt (a felbontással és a magnyelvekkel kapcsolatos) fogalmak és tételek önmagukban is érdekesek, és velük összefüggésben további matematikai problémák is fölvetődnek. A magnyelvek vonatkozásában elsősorban az a kérdés merül föl, hogy az egyes magnyelvek szerkezete (amely *sematikusan tükrözi a programok viselkedését*) hogyan függ az őket „előállító" programok logikai szerkezetétől, s a magnyelvek osztálya hogyan viszonyul a formális nyelvek *Chomsky-féle hierarchiájához?* A jelen dolgozatban ebben az irányban végzünk vizsgálatokat, mégpedig az ún. *egyszerű számlálásos ciklusokból felépülő programok* magnyelveivel kapcsolatban. (Az ilyen programok pontosan azok, amelyek nemnegatív egész értékekkel dolgoznak, és amelyek fokozatosan úgy építhetők föl az értékadó utasításaikból kiindulva, hogy egy-egy lépésben csak *soros kompozíciót* vagy pedig *„egyszerű számlálásos ciklusba foglalást"* végezhetünk; a program működése során egy-egy ilyen ciklusban a ciklusfej minden átfutásakor 1-gyel csökkenti a ciklusváltozót, amíg az 0-vá nem válik, a ciklustörzs azonban nem módosíthatja a ciklusváltozó értékét.)

Az egyszerű számlálásos ciklusokból felépíthető programok lényegében megfelelnek MEYER és RITCHIE [8] *„loop-programjainak"*, amelyek pontosan a primitív rekurzív függvényeket tudják kiszámítani (l. [2, 4, 8]). Általánosabban belátható ugyanez az egyszerű számlálásos ciklusokból felépíthető programokról is (amennyi-

ben pl. a bennük lehetséges értékadó utasítások a közvetlen érték-átadást, valamint a nemnegatív egészeken végezhető bármelyik aritmetikai művelet, ill. a hatványozás eredményének értékül adását jelenthetik). Mi indokolja, hogy dolgozatunkban — a nemnegatív egészekkel dolgozó — tetszőleges programok helyett, ezeknek csak egy speciális részosztályát, az egyszerű számlálásos ciklusokból felépíthető programokét, vesszük alapul a programok magnyelveinek vizsgálata során? Az, hogy amint erre még visszatérünk, BRAINERD és LANDWEBER [2] és SCHNORR [11] könyvéből ismeretes, hogy a rekurzív függvények közül azok, amelyek gyakorlatilag elfogadható időn belül kiszámíthatóak (bármilyen korszerű, nagyteljesítményű számítógépen), nemcsak mind primitív rekurzívok, hanem még ez utóbbi függvényeknek is egy szűk részosztályában a *Kalmár-féle elemi függvények* (l. [1, 2, 5, 7, 11]) osztályában vannak. Az egyszerű számlálásos ciklusos programok tehát a gyakorlat igényeit még „túlzottan is" kielégítik.

Vizsgálatainkban a konkrét programozási nyelvek helyett egy leegyszerűsített absztrakt programnyelvet használunk: az *„URM assembly nyelvet"*. Ez a nyelv a SHEPHERDSON és STURGIS [12] cikkében definiált és későbbi szerzők [2, 4, 11] által egyszerűsített absztrakt *URM gép* programjainak áttekinthető írására szolgál. (Az ezen a nyelven írt egyszerű számlálásos ciklusos programok egyébként majdnem egybeesnek MEYER és RITCHIE [8] *loop programjaival.*) Az URM assembly nyelvet csupán egyszerűsége miatt választottuk, és ez sem jelent lényeges elvi megszorítást, két okból sem. Egyrészt (mint fentebb már említettük), az egyszerű számlálásos ciklusos programok által kiszámítható függvények osztálya (amely éppen a primitív rekurzív függvények osztálya, l. még ehhez [2, 4, 8]-at) nem bővül, ha az URM assembly nyelv értékadó utasításai helyett valamilyen konkrét, magasabb szintű nyelvet (pl. a FORTRAN-ét vagy az ALGOL-ét) használjuk; másrészt, az egyszerű számlálásos ciklusos programok „lefutási módja" (amelyet a programok magnyelvei tükröznek, s amelynek tanulmányozásában ezért a magnyelvek jól felhasználhatóak) a programgráf szerkezetétől függ, nem pedig a konkrét programnyelvtől.

Rátérünk most az URM assembly nyelv definiálására. Jelöljük a változókat és címkéket olyan jelsorozatokkal, melyek latin kisbetűvel kezdődnek, utána további latin kisbetűk és számjegyek következhetnek. A változók nemnegatív egész értékűek. A nyelv utasításai és jelentéseik a következők:

az utasítás:	és jelentése:
(c:) IC x	$x:=x+1$
(c:) DC x	$x:=\begin{cases} x-1, & \text{ha } x\geqq 1 \\ 0, & \text{ha } x=0 \end{cases}$
(c:) RW x, y	$x:=y$ (és y eredeti értéke megmarad)
(c:) JP d	*menj* a d címkéjű utasításra
(c:) BR x, d	*ha* $x=0$, *akkor menj* a d címkéjű utasításra, *különben menj* a program következő utasítására.

(Itt x és y tetszőleges változó, c és d tetszőleges címke, a címkék zárójelbe tevése azt jelöli, hogy címkét nem kötelező kitenni, ha a programban nincs hivatkozás az illető helyre. Természetesen, egy címke csak egy helyet jelölhet.)

Egy URM (assembly nyelvű) program ezekből az utasításokból épül fel. A program végrehajtása az első utasítás végrehajtásával kezdődik, HALT utasítása nincs a nyelvnek, ezért a végrehajtás akkor fejeződik be, ha a programban nemlétező címkére kellene ugrani vagy elágazni, vagy az utolsó helyen álló utasítás végrehajtása után, a rákövetkező, nemlétező utasítást kellene végrehajtani.

A programba mindig beleértjük, hogy meg van hozzá adva a bemenő változóinak halmaza is (s az egyszerűség kedvéért, és a magnyelv általunk megadandó, módosított definíciójával összhangban, a program minden változóját kimenő változónak tekintjük). A program nem-bemenő változói a segédváltozók (tehát az összes bemenő- és segédváltozó egyúttal kimenő változó is). Megállapodunk abban, hogy a program végrehajtásának kezdetén a bemenő változók értékei tetszőlegesek (nemnegatív egészek), a segédváltozóké pedig 0. Abban is megállapodunk, hogy programok összefűzésekor az egyes bemenő-változó halmazok egyesítése lesz az „eredő" program bemenő változóinak halmaza.

DEFINÍCIÓ: Legyen P tetszőleges URM-program, és tegyük fel, hogy az x változó nem szerepel P-ben. Ekkor P egyszerű számlálásos ciklusba foglalása jelentse a következő P′ program megszerkesztését (föltesszük, hogy P-ben nem szerepel a $c1$ és $c2$ címke):

$c1$: BR x, $c2$
DC x
P
JP $c1$
$c2$:

A P' végrehajtása során tehát P-t annyiszor hajtjuk végre, amennyi az x kezdeti értéke. (A $c2$ címke a JP $c1$ utasítás utáni — üres — hely címe.)

Ezután induktívan definiálhatjuk az egyszerű számlálásos ciklusokból felépülő programot (itt most az egyszerűbb írásmód kedvéért föltesszük, hogy minden programban, az i-edik helyen álló utasítás címkéje ci, és soros kompozíció és egyszerű számlálásos ciklusba foglalás során ennek megfelelő átcímkézést hajtunk végre):

$\mathscr{SC}(0)$ legyen az olyan nem-üres URM-programok halmaza, amelyek nem tartalmaznak JP és BR utasítást (így az $\mathscr{SC}(0)$-beli programokban nincs ciklus).

Most föltéve, hogy valamely $i \in N$-re $\mathscr{SC}(i)$-t már definiáltuk, $\mathscr{SC}(i+1)$ legyen a következő:

$\mathscr{SC}(i+1) := (\mathscr{SC}(i) \cup \{\text{egyszerű számlálásos ciklusba foglalt } P \text{ program } | P \in \mathscr{SC}(i)\})^+$ (megfelelő átcímkézéssel, ld. a föntebbi megállapodást).

Tehát $\mathscr{SC}(i+1)$ olyan URM-programok halmaza, melyekben az IC, DC, RW utasítások, és a ciklusok kialakításában a BR és JP utasítások szerepelnek, a ciklusok egymásba ágyazásának mélysége legföljebb $(i+1)$.

Az $\mathscr{SC}(i)$-beli programokkal kiszámítható függvények osztályát $SC(i)$-vel jelöljük ($i \in N$). Az $SC(i)$ ($i \in N$) osztályok egyesítése pedig legyen SC:

$$SC := \bigcup_{i \in N} SC(i)$$

(s ez éppen a primitív rekurzív függvények osztálya, l. [2, 4, 8]).

Visszatérünk most a függvények gyakorlati kiszámíthatóságával kapcsolatos korábbi megjegyzésünkre. MEYER és RITCHIE [8], BRAINERD és LANDWEBER [2] és SCHNORR [11] munkájából ismeretes, hogy az SC(2)-beli függvények pontosan azok, amelyeknek van olyan (tetszőleges, nem föltétlenül számlálásos ciklusos) P URM programjuk, és ehhez olyan $m \in N$, hogy P lépésszáma, $t_P < f^{(m)}$, ahol $f(x) = 2^x$ (azaz z bemenő vektorra,

$t_P(z) < f^{(m)}(\max(z)) = 2^{2^{\cdot^{\cdot^{\cdot^{2^{\max(z)}}}}}}$ (m db), ahol $\max(z)$ a z vektor maximális komponense).

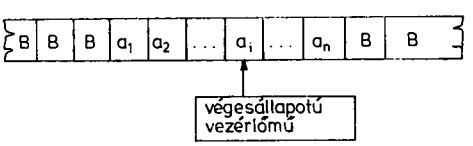
Így, ha egy függvény nem esik ezen függvényosztályba, akkor lépésszáma végtelen sok helyen olyan nagy, hogy ezáltal gyakorlatilag kiszámíthatatlan.

Egyébként BRAINERD és LANDWEBER könyvéből [2] az is ismeretes, hogy az $SC(2)$ megegyezik a *Kalmár-féle elemi függvények* osztályával (KALMÁR [7], BERECZKI [1], GRZEGORCZYK [5]).

2. A formális nyelvek elméletéből használt fogalmak és tételek

Ebben a fejezetben a formális nyelvek és automaták elméletéből néhány olyan definíciót és tételt ismertetünk, amelyekre a továbbiakban szükségünk lesz.

A *nemdeterminisztikus Turing-gép* definíciójának több változata is szerepel a szakirodalomban, s ezekről bebizonyítható, hogy egyenértékűek (HARRISON [6], RÉVÉSZ [10], VARGA [14]). Ezért definiáljuk azt a változatot, amit használni fogunk.



1. ábra

A *Turing-gép* legegyszerűbb változata egy véges állapotú vezérlőműből, egy író-olvasó fejből, és egy kétirányban végtelen hosszúságú szalagból áll, amely „mezőkre" van felosztva (1. ábra).

A bemenő szót, jelenként, annyi egymásutáni mezőbe írjuk be, amennyi jelből áll, s az így felírt bemenő szótól jobbra és balra minden mezőben a B-vel jelölt „üresjel" vagy „blankjel" található (a B jel mindegyik a_i-től különbözik). A *Turing-gépet* a bemenő szó első jelét tartalmazó mezőből indítjuk el (üres bemenő szó esetén tetszőleges mezőből), kezdőállapottal.

A *Turing-gép* minden egyes lépése a következő műveletek végrehajtásából áll:
— beolvassa a szalagról az író-olvasó fej által éppen vizsgált mezőben levő jelet,
— megváltoztatja a vezérlőmű állapotát,
— kiír egy új jelet (az eredeti helyett) az éppen vizsgált mezőbe,
— egy mezővel jobbra vagy balra lépteti az író-olvasó fejet, vagy helyben hagyja.

Formálisan, a gép működését olyan (q, x, p, y, d) rendezett ötösök halmazával lehet leírni, ahol q a vezérlőmű pillanatnyi állapota, x az író-olvasó fej által éppen vizsgált jel, p a megfelelő új állapot, y a kiírt jel, $d \in \{R, L, H\}$, ahol az R, L, illetve H a jobbra lépést, balra lépést, illetve a helyben maradást jelenti.

2.1. DEFINÍCIÓ: Egy *nemdeterminisztikus Turing-gép* olyan $A = (Z, \Sigma, K, M, q_0, F)$ rendezett hatos, ahol

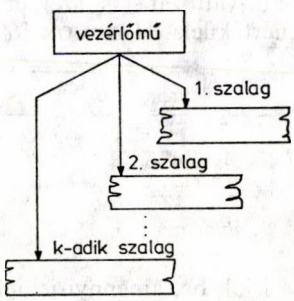
Z: a (véges, nem-üres) szalag-ábécé,
Σ: a (nem-üres) bemenő ábécé, $\Sigma \subset Z$, $B \in Z \setminus \Sigma$,
K: a (véges, nem-üres) állapothalmaz,
$q_0 \in K$: a kezdőállapot,
$F \subseteq K$: a végállapotok halmaza,
M: a $K \times Z$ halmaznak egy leképezése a $K \times Z \times \{R, L, H\}$ hatványhalmazába: átmenetfüggvény.

Az M leképezést a következőképpen fogjuk megadni: $(q, y) \to (p, z, d)$, ha $(p, z, d) \in M(q, y)$.

A nemdeterminisztikusság azt jelenti, hogy valamely (q, y)-hoz tetszőleges véges (esetleg zérus) számú (p, z, d) alakú hármas tartozhat.

2.2. DEFINÍCIÓ: Ha az *A nemdeterminisztikus Turing-gép,* miután a bemenő szó minden jelét legalább egyszer megvizsgálta, a működése során valamelyik végállapotába jut, akkor (és csak akkor) azt mondjuk, hogy az A elfogadta a (Σ^*-beli) bemenő szót. Az A által elfogadott nyelv, $L(A) :=$ az A által elfogadott bemenő szavak halmaza.

2.3. DEFINÍCIÓ: A lineárisan korlátos automata olyan *nemdeterminisztikus Turing-gép*, amely a szalagjából legföljebb olyan hosszú darabot (annyi mezőt) használhat, mint a bemenő szó hosszának valamely lineáris függvénye. Azaz, létezik olyan $c_1 > 0$ és $c_2 \geqq 0$ szám, hogy tetszőleges w bemenő szó esetén, a szalagon felhasznált mezők száma nem több, mint $c_1|w| + c_2$.



2. ábra

2.1. TÉTEL: A lineárisan korlátos automaták által elfogadott nyelvek osztálya azonos a környezetfüggő nyelvek osztályával.

Sokszor megkönnyíti a munkánkat, ha föltesszük, hogy a *Turing-gépünknek* nem egyetlen szalagja van, hanem például k darab, melyek mindegyikéhez tartozik egy-egy író-olvasó fej, és minden egyes lépés a k darab szalag mindegyikén párhuzamosan egy-egy olyan műveletnek a végrehajtását jelenti, amilyent az *egyszalagú Turing-gépnél* már láttunk. Az 1. szalag bemenő- és munkaszalagként, a 2.-ik, ..., k-adik szalag pedig további munkaszalagként szolgál. A működés kezdetén a munkaszalagokon csupa B jel van. A 2. ábra egy *k-szalagú Turing-gépet* mutat.

2.4. DEFINÍCIÓ: Egy *k-szalagú nemdeterminisztikus Turing-gép* olyan

$A = (Z_1, \ldots, Z_k, \Sigma, K, M, q_0, F)$ rendezett $(k+5)$-ös, ahol
Z_i: az i-edik szalag (véges, nem-üres) ábécéje, $B \in Z_1 \cap \ldots \cap Z_k$
Σ: a (nem-üres) bemenő ábécé, $\Sigma \subset Z_1$, $B \notin \Sigma$,
K: a (véges, nem-üres) állapothalmaz,
$q_0 \in K$: a kezdőállapot,
$F \subseteq K$: a végállapotok halmaza,
M: a $K \times Z_1 \times \ldots \times Z_k$ halmaznak egy leképezése a $K \times Z_1 \times \ldots \times Z_k \times \{R, L, H\}^k$ hatványhalmazába: átmenetfüggvény.

Eszerint, ha $p, q \in K$, $x_i \in Z_i$, $y_i \in Z_i$ és $d_i \in \{R, L, H\}$ $(1 \leqq i \leqq k)$, akkor a

$$(p, y_1, \ldots, y_k; d_1, \ldots, d_k) \in M(q, x_1, \ldots, x_k)$$

összefüggés azt jelenti, hogy a *k-szalagú nemdeterminisztikus Turing-gép*, ha a q állapotban van és x_i-t olvas be az i-edik szalagjáról $(i=1, \ldots, k)$, akkor átmehet a p állapotba úgy, hogy az i-edik szalagra y_i-t ír, majd az i-edik szalagon egy mezővel jobbra vagy balra lép, vagy helyben marad, aszerint, hogy $d_i = R, L$ vagy H.

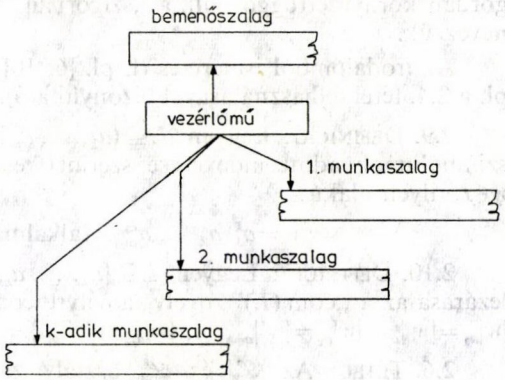
A *többszalagú nemdeterminisztikus Turing-gépek* által elfogadott szavakat és nyelveket szintén a 2.2. Definíció szerint értelmezzük.

2.5. Definíció: Egy *többszalagú nemdeterminisztikus Turing-gép* akkor többszalagú lineárisan korlátos automata, ha valamennyi szalagjának a működés során felhasznált része nem hosszabb, mint a bemenő szó hosszának valamely lineáris függvénye.

Bármely többszalagú lineárisan korlátos automata szimulálható egyszalagúval, ezért a 2.1. tétel alapján igaz a következő,

2.2. Tétel: A többszalagú lineárisan korlátos automaták által elfogadott nyelvek osztálya megegyezik a környezetfüggő nyelvek osztályával.

Mi a többszalagú lineárisan korlátos automatának egy olyan speciális változatát fogjuk használni, amelyben a bemenő szó egy külön bemenőszalagon van. Ez a szalag kizárólag a bemenő szó bevitelére szolgál, a mezők tartalmát az automata csak beolvasni tudja, megváltoztatni nem. A bemenőszalag csak egyirányban (jobbfelé) végtelen, a bemenő szó e szalag valamely (véges) kezdőszeletét foglalja el (a többi mezőben B jel van), s a bemenőszalagon az író-olvasó fej kezdetben az első mezőre mutat. Van még az automatának k db ún. munkaszalagja (ezek kétirányban végtelenek, és kezdetben csupa B jelet tartalmaznak). A bemenőszalag elejére írt valamely w szót akkor fogadja el az automata, ha ezen a szalagon közvetlenül a w utáni mezőre jut végállapotban. Föltehetjük azt is, hogy az automatának egyetlen végállapota van (ha a végállapotok száma 1-től eltérő, akkor az automata által elfogadott nyelv megváltoztatása nélkül úgy módosíthatjuk az automatát, hogy egyetlen végállapota legyen). (Természetesen, a *többszalagú nemdeterminisztikus Turing-gépek* esetében is mindig elegendő csupán az előbbiekhez hasonló, speciális változatokat figyelembe vennünk.) A többszalagú lineárisan korlátos automatának a most leírt speciális változatát *t.l.k. automatának* fogjuk nevezni (a „többszalagú lineárisan korlátos automata" rövidítéseként). Ezt az automatatípust a 3. ábra szemlélteti. Valamely A *t.l.k. automata* által elfogadott nyelv jele: $L(A)$. Világos, hogy a *t.l.k. automatákra* is igaz a 2.2. tétel, azaz:



3. ábra

2.3. TÉTEL: A *t.l.k. automaták* által elfogadott nyelvek osztálya azonos a környezetfüggő nyelvek osztályával.

A következő tétel egy olyan föltételt határoz meg, amely szükséges (de nem elégséges) ahhoz, hogy egy nyelv környezetfüggetlen legyen.

2.4. TÉTEL (BAR—HILLEL LEMMA): Bármely L környezetfüggetlen nyelvhez megadható két természetes szám, p és q úgy, hogy minden olyan $P \in L$ szó, amelyre $|P| > p$, felírható $P = uxwyz$ alakban oly módon, hogy $|xwy| \leqq q$, $xy \neq \lambda$ és minden $i \in N$-re $ux^i wy^i z \in L$.

2.6. DEFINÍCIÓ: Veremautomatának az olyan $A = (Z, K, T, M, z_0, q_0, H)$ rendezett hetest nevezzük, amelyben
Z véges, nem-üres ábécé: a verem-ábécé,
K véges, nem-üres halmaz: az állapothalmaz,
T véges, nem-üres ábécé: a bemenő ábécé,
M leképezés a $Z \times K \times (T \cup \{\lambda\})$ halmazról a $Z^* \times K$ véges részhalmazainak halmazába: az átmenetfüggvény,
$z_0 \in Z$ a kezdőjel,
$q_0 \in K$ a kezdőállapot,
$H \subseteqq K$ a végállapotok halmaza.

2.7. DEFINÍCIÓ: Egy A veremautomata egy $X \in Z^*KT^*$ szót egy lépésben átalakít az $Y \in Z^*KT^*$ szóba (jelben $X \underset{A}{\Rightarrow} Y$), ha van olyan $z \in Z$, $q, p \in K$, $a \in T$, valamint $w, W \in Z^*$, $P \in T^*$, melyekre $(w, p) \in M(z, q, a)$ és $X = WzqaP$, $Y = WwpP$ vagy $(w, p) \in M(z, q, \lambda)$ és $X = WzqaP$, $Y = WwpaP$. Egy A veremautomata az X szót véges sok lépésben átalakítja az Y szóba (jelben $X \overset{*}{\underset{A}{\Rightarrow}} Y$), ha $X = Y$, vagy van olyan $X_1, X_2, \ldots, X_n$ sorozat, melyre $X = X_1$, $Y = X_n$ és $X_i \underset{A}{\Rightarrow} X_{i+1}$ $(i = 1, 2, \ldots, n-1)$. Egy A veremautomata által végállapottal elfogadott nyelv a következő:

$$L(A) := \left\{P \in T^* \mid z_0 q_0 P \overset{*}{\underset{A}{\Longrightarrow}} Wp, \text{ valamely } W \in Z^*\text{-ra és } p \in H\text{-ra}\right\}.$$

2.5. TÉTEL: A veremautomaták által végállapottal elfogadott nyelvek osztálya azonos a környezetfüggetlen nyelvek osztályával.

2.8. DEFINÍCIÓ: Az $\hat{\mathscr{L}}_1 := \mathscr{L}_1 \setminus \mathscr{L}_2$ és az $\hat{\mathscr{L}}_2 := \mathscr{L}_2 \setminus \mathscr{L}_3$ nyelvosztályt a „szigorúan környezetfüggő", ill. a „szigorúan környezetfüggetlen" nyelvek osztályának nevezzük.

Az irodalomból ismeretes (l. pl. [6, 10]), hogy $\hat{\mathscr{L}}_1 \neq \varnothing$ és $\hat{\mathscr{L}}_2 \neq \varnothing$ (az előbbi pl. a 2.4. tétel felhasználásával bizonyítható).

2.9. DEFINÍCIÓ: Legyen $L \subseteqq \{a_1, \ldots, a_m\}^*$ tetszőleges nyelv. L-et (az $a_1, \ldots, a_m$ szimbólumok adott indexezése szerint) rendezett nyelvnek nevezzük, ha minden $w \in L$ ilyen alakú:

$$w = a_1^{k_1} a_2^{k_2} \ldots a_m^{k_m} \quad \text{(alkalmas } k_1, \ldots, k_m \in N\text{-re)}.$$

2.10. DEFINÍCIÓ: Legyen $L \subseteqq \{a_1, \ldots, a_m\}^*$ tetszőleges nyelv. Az L kommutatív lezárása az a „com (L)" nyelv, amelyre com $(L) := \{w \mid w \in \{a_1, \ldots, a_m\}^*$, és $\exists w' \in L$: $|w|_{a_1} = |w'|_{a_1}$, $|w|_{a_2} = |w'|_{a_2}$, $\ldots$, $|w|_{a_m} = |w'|_{a_m}\}$.

2.6. TÉTEL: Az $\mathscr{L}_2$ és $\mathscr{L}_3$ osztály zárt tetszőleges reguláris nyelvvel való metszésre.

3. Az egyszerű számlálásos ciklusokból felépíthető programok magnyelveinek környezetfüggő volta

Az egyszerű számlálásos ciklusokból felépíthető programok magnyelveit vizsgáljuk.

Definiáljuk először a *Pratt-féle felbontást* az egyszerű számlálásos ciklusokból felépíthető URM-programokra.

3.1. DEFINÍCIÓ: A JP és BR utasítások a vezérlő részhez tartoznak. Minden RW, IC és DC (értékadó) utasítás legyen magutasítás, s amelyikük vezérlési funkciót is ellát (pl. egy későbbi ciklusváltozót növel), az legyen egyúttal vezérlő utasítás is.

Valamely „$x:=\ldots$" utasítás akkor vezérlő utasítás, ha

— vagy ezen utasítástól vezet egy út egy „BR x, c" alakú utasításhoz, és ezen az úton nincs több értékadás x-re (vagyis egy olyan ciklushoz, amelynek x a ciklusváltozója),

— vagy ezen utasítástól vezet egy út egy másik vezérlő értékadó utasításhoz, amely (a „jobb oldali kifejezésében") hivatkozik x-re, és az úton nincs több értékadás x-re.

A vezérlő utasításokban szereplő változókat nevezzük vezérlő változóknak.

Nyilvánvaló, hogy minden utasítás hozzátartozik vagy a vezérlő-, vagy a magrészhez, és lehetnek olyan utasítások is, amelyek mindkettőhöz.

A fenti definícióban PRATT [9] cikkétől egy kissé eltérve, a magrészhez minden értékadást hozzávettünk, hogy a kapott kiszámítási sorozat a számítás teljes menetét tükrözze. Ilyenkor ugyanis minden ciklus-végrehajtás megjelenik a sorozatban, hiszen minden ciklusban a ciklusváltozó csökken (DC utasítás). Az ellenkező esetben lehetnének olyan ciklusok, amelyek nem szerepelnének a kiszámítási sorozatban, ekkor a sorozatból a program végrehajtásának csak egy részét lehetne nyomon követni. Az ilyen magnyelvet is érdekes lehet formális nyelvi szempontból vizsgálni. Ekkor azonban sokkal bonyolultabb is lehet a magnyelv. Megmutatható ugyanis, hogy ilyen esetben előállhat nem-környezetfüggő, sőt nem-rekurzív magnyelv is.

A magnyelv fenti definíciója mellett vizsgáljuk meg, milyen kapcsolatban vannak az egyszerű számlálásos ciklusokból felépíthető programok magnyelvei a nyelvek CHOMSKY által definiált osztályozásával.

Vezessük be a következő jelöléseket:

$\mathscr{C}_i$ legyen az $\mathscr{SC}(i)$-beli programok magnyelveinek osztálya,

$\mathscr{C}_{i,j}$ legyen az olyan $\mathscr{SC}(i)$-beli programok magnyelveinek osztálya, amelyekben j darab i-mélységű ciklus van.

Így

$$\mathscr{C}_{0,0} = \emptyset,$$

$$\mathscr{C}_i = \bigcup_{j \in N_+} \mathscr{C}_{i,j},$$

és $(i \in N)$

$$\mathscr{C}_{i+1,0} = \mathscr{C}_i.$$

Könnyű belátni, hogy tetszőleges $\mathscr{SC}(0)$-beli program magnyelve reguláris. Ezek a programok ugyanis csak IC i, DC j és RW k, l alakú utasításokból állnak, melyek a leírás sorrendjében kerülnek végrehajtásra, mindegyik pontosan egyszer. Így minden ilyen program magnyelve egyetlen magszimbólumból áll (s egy ilyen program „0-mélységű ciklusnak" tekinthető).

Az $\mathscr{SC}(1)$-beli programoknál először azt nézzük meg, ha egyetlen ciklus van a programban. A program ekkor a következő:

P_1

$c1$: BR x, $c2$

DC x

P_2

JP $c1$

$c2$: P_3

ahol $P_1, P_2, P_3 \in \mathscr{SC}(0)$, és a $P_1, (\mathrm{DC}\, x) P_2$, ill. P_3 programrészeknek megfelelő magszimbólumok legyenek K_1, K_2, ill. K_3. E program magnyelve: $\{K_1 K_2^n K_3 | n$ a ciklus valamely végrehajtási száma$\}$. Az n értéke a ciklusváltozó kezdeti értékétől függ, amit IC, DC és RW utasítás módosíthat. Ha a P_1 programrészben értékadó utasítás nem változtatja az x ciklusváltozó értékét, akkor a magnyelv: $\{K_1 K_2^* K_3\}$, ha x bemenő változó, és $\{K_1 K_3\}$ különben.

Ha P_1-ben DC és IC utasítások előfordulnak, és x bemenő változó, akkor ezek számától és sorrendjétől függően x-re a ciklus elején az $x \geqq k$ feltétel igaz valamilyen konstans $0 \leqq k$-ra, és a magnyelv ilyenkor $\{K_1 K_2^k K_2^* K_3\}$. Pl. csak DC utasítások előfordulása esetén $k=0$, hiszen a DC utasítás csak $x>0$ esetben csökkenti x értékét, tehát x a ciklus kezdetén ugyanúgy befutja az N_+ tartományt, mint az első esetben. Ha x segédváltozó, akkor a magnyelv nyilván $\{K_1 K_2^k K_3\}$.

Ha P_1-ben RW x, y utasítás is előfordul, akkor a fenti gondolatmenet alkalmazható úgy, hogy az RW utasítás előtt y-t, utána pedig x-et vizsgáljuk, és a magnyelv nyilván nem változik.

Összességében tehát a vizsgált programokhoz tartozó magnyelvek $\{K_1 K_2^k K_3\}$ vagy $\{K_1 K_2^k K_2^* K_3\}$ alakúak ($k \in N$) lehetnek, ezek a nyelvek mind regulárisak. Vagyis, $\mathscr{C}_{1,1} \subseteq \mathscr{L}_3$.

Ha 2 ciklus van a programban:

P_1

$c1$: BR x, $c2$

DC x

P_2

JP $c1$

$c2$: P_3

$c3$: BR y, $c4$

DC y

P_4

JP $c3$

$c4$: P_5,

ahol $P_1, P_2, P_3, P_4, P_5 \in \mathscr{SC}(0)$, a $P_1, (\mathrm{DC}\ x)P_2, P_3, (\mathrm{DC}\ y)P_4$ és P_5 programrészeknek megfelelő magszimbólumok pedig rendre K_1, K_2, K_3, K_4 és K_5. E program magnyelve $K_1K_2^{n_1}K_3K_4^{n_2}K_5$ alakú szavakból áll, amelyekben n_1 és n_2 az első és a második ciklus összetartozó végrehajtási számai. Például a $\{K_2^nK_4^n | n \in N\}$ nyelv egy ilyen program magnyelve. Erről a nyelvről az irodalomból tudjuk, hogy „szigorúan" környezetfüggetlen (vagyis, $\in \hat{\mathscr{L}}_2$). Tehát két ciklus esetén kilépünk a reguláris nyelvek osztályából, de könnyű belátni, hogy a környezetfüggetlen osztályban maradunk, azaz $\mathscr{C}_{1,2} \subseteq \mathscr{L}_2$. (Ha a két ciklus ciklusváltozója független egymástól, akkor reguláris a magnyelv; ha nem függetlenek, akkor n_2 valamilyen lineáris függvénye n_1-nek; az előbbi példában ennek speciális eseteként, $n_2 = n_1 = n \in N$ teljesült.)

Ha három ciklus van a (P-vel jelölt) programban:

P_1

$c1$: BR x, $c2$

DC x

P_2

JP $c1$

$c2$: P_3

$c3$: BR y, $c4$

DC y

P_4

JP $c3$

$c4$: P_5

$c5$: BR z, $c6$

DC z

P_6

JP $c5$

$c6$: P_7

ahol $P_1, \ldots, P_7 \in \mathscr{SC}(0)$, a $P_1, (\mathrm{DC}\ x)P_2, P_3, (\mathrm{DC}\ y)P_4, P_5, (\mathrm{DC}\ z)P_6$ és P_7 programrészeknek megfelelő magszimbólumok legyenek $K_1, \ldots$ és K_7.

Az $L(P)$ magnyelv $K_1K_2^{n_1}K_3K_4^{n_2}K_5K_6^{n_3}K_7$ alakú szavakból áll. Ha legföljebb két ciklusváltozó értéke függ csak egymástól, akkor megegyezik az eddig vizsgált esetekkel.

Ha mindhárom ciklusváltozó értéke függ egymástól, akkor azonban kilépünk a környezetfüggetlen nyelvek osztályából. Nézzük meg például a $\{K_2^nK_4^nK_6^n | n \in N\}$ nyelvet. Erre alkalmazva a *Bar—Hillel-lemmát* (2.4. tétel), könnyen megmutatható, hogy nem környezetfüggetlen. De az is könnyen belátható, hogy ez a nyelv környezetfüggő.

Az általános esetben ennél jóval bonyolultabb nyelveket kaphatunk. Vajon ezek is mind környezetfüggőek lesznek-e, vagy vannak-e olyan $\mathscr{SC}(1)$-beli programok, amelyeknek magnyelvei ebből a nyelvosztályból is kilépnek?

Tudjuk, hogy egy nyelv akkor környezetfüggő, ha készíthetünk hozzá olyan *t.l.k. automatát,* amely elfogadja. Készítsünk ilyen automatát a föntebbi $L(P)$ nyelvhez. Az automata bemenő szalagján legyen a nyelv aktuális szava. A ciklusváltozók értékeit pedig számolja az automata egy-egy külön munkaszalagon.

Ennek biztosításához minden vezérlő változóhoz rendelünk egy-egy munkaszalagot. A bemenőszalagra nem ír, így hosszát nem növeli, a munkaszalagokon viszont a ciklusváltozók értékeit úgy számolja, hogy amennyi éppen a ciklusváltozó értéke, annyi egyes van a neki megfelelő munkaszalagon. Így a lineárisan korlátos automatánál szükség van arra, hogy ezen szalagok hosszát a bemenő szó hosszának valamely lineáris függvényével felülről becsülni tudjuk. (A változók értékei csak a vezérlés szempontjából érdekesek, ezért elég őket csak a vezérlő utasításokban felülről becsülni, és csak a vezérlő értékadásokat figyelni.)

3.1. LEMMA: Minden L magnyelvhez megadható olyan $c>0$ egész szám, hogy tetszőleges $w \in L$-re a vezérlő változóknak a vezérlő utasításokban a w-hez tartozó kiszámítás során fellépő értékei nem nagyobbak, mint $c \cdot |w|$.

BIZONYÍTÁS: Válasszunk ki egy tetszőleges x ciklusváltozót, ami w-nél valóban ciklusváltozó. Az utolsó olyan K_i magszimbólum után, amely e ciklus magja, a ciklusváltozó értéke 0.

Innen előre és hátra haladva becsüljük x értékét:

— előre: mivel felső becslés kell, csak a vezérlő DC x utasítás érdekes, mivel ebben az esetben előtte nagyobb volt az x értéke, ezért ott a mostani becslés nem jó. Minden magszimbólumhoz meghatározott utasítások (véges sok) tartoznak, így van olyan c_1 konstans, hogy előrehaladva minden szimbólumhoz legföljebb c_1 számú DC x utasítás tartozik. Így e rész hosszának c_1-szerese jó felső becslés, s ez „még inkább" igaz lesz $c_1 \cdot |w|$-re.

— hátra: a felső becslés miatt itt csak a vezérlő IC x utasítás érdekes, mivel utána több lesz a ciklusváltozó értéke, így a becslést is növelni kell. Az előző esethez hasonlóan itt is van olyan $c_2>0$ konstans, hogy hátrafelé haladva minden szimbólumhoz legföljebb c_2 számú IC x utasítás tartozik. Így $c_2 \cdot |w|$ jó lesz becslésnek e részre.

Az egész programban a felső becslés x-re ekkor a következő lesz:

$$\max(c_1 |w|, c_2 |w|) = \max(c_1, c_2) \cdot |w|$$

$$\tilde{c} := \max(c_1, c_2).$$

Minden ciklusváltozóra kiszámíthatjuk $\tilde{c}$ értékét. A nem-ciklusváltozók értékének becslését egy RW utasításnál kezdjük, ahol ez a változó egy ciklusváltozónak ad értéket. (Itt a ciklusváltozóra van becslés.) Ebből kiindulva előre és hátra haladva az előbbi módon végezzük a becslést ($\tilde{c}$). Végül a $\tilde{c}$-ok maximumát véve c-nek, az állítás igaz lesz.

Az automatának az lesz a feladata, hogy figyelje, K_i után a megfelelő szimbólum következik-e, és a ciklusok annyiszor hajtódnak-e végre, amennyi a munkaszalagokon számolt érték.

Ehhez első lépésként nem-determinisztikusan kezdeti értékeket generál a bemenő változók szalagjain: a bemenő szalagon az író-olvasó fej egyesével jobbra lép a szimbólumokon. Minden egyes szimbólumnál legföljebb (a lemmában szereplő) c darab egyest írhat minden külön szalagra. Majd visszalép az első szimbólumra.

Az automata, működése közben, kell, hogy szimulálja a program vezérlő értékadásait a külön szalagokon, hogy ellenőrizni tudja, annyiszor hajtódott-e végre egy-egy ciklus, ahányszor kellett.

Ezért egyesével halad az író-olvasó fej jobbra a szimbólumokon, és minden szimbólumnál a megfelelő vezérlő értékadásokat „elvégzi" a munkaszalagokon. Ezek az értékadások

a) IC x

b) DC x

c) RW x, y

utasításokkal történhetnek, ahol x és y vezérlő változók.

Először nézzük meg, hogy az a), b) és c) utasítások hogyan szimulálhatók a munkaszalagokon.

Tegyük föl, hogy valamely p állapotban van az automata, a közbülső állapotokat csak itt veszi fel.

a) Ha IC k utasítás volt, akkor a k változóhoz rendelt szalagon egy egyest ír az utolsó egyes (vagy B jel) után, és eggyel jobbra lép:

$$(p, K, \ldots, B, \ldots) \to (p', K, \ldots, 1, \ldots; H, \ldots, R, H, \ldots)$$

a bemenő szalagon — a k változóhoz rendelt szalagon

(megjegyzés: a pontok a többi szalagra vonatkoznak, azokat nem változtatja).

b) A DC k utasításnál meg kell különböztetni azt az esetet, amikor k éppen ciklusváltozó (K annak a ciklusnak a magjához tartozik, amelynek k a ciklusváltozója), ugyanis ekkor nem fordulhat elő, hogy k értéke 0 (a szalagja üres) és a bemenő szalagon az író-olvasó fej K-ra mutat;
— ha k nem ciklusváltozó: az automata a k változó szalagján az utolsó egyest, ha volt, törli (előbb visszalép egyet):

$$(p, K, \ldots, B, \ldots) \to (p_2', K, \ldots, B, \ldots; H, \ldots, L, \ldots),$$

ekkor, ha volt egyes:

$$(p_2', K, \ldots, 1, \ldots) \to (p', K, \ldots, B, \ldots; H, \ldots, H, \ldots),$$

ha nem volt egyes:

$$(p_2', K, \ldots, B, \ldots) \to (p', K, \ldots, B, \ldots; H, \ldots, H, \ldots),$$

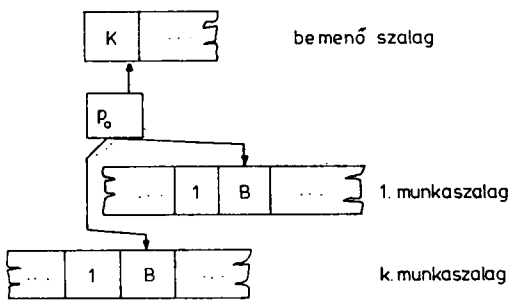
— ha k ciklusváltozó: az utóbbi lépést nem értelmezi.

c) Ha RW k, l utasítás volt: az l változó szalagját át kell másolni a k változó szalagjára. Az automata viselkedése az előzőekhez hasonlóan írható le.

Minden magszimbólum egy meghatározott programrészhez tartozik, amiben több ilyen vezérlő értékadás is lehet. Ezért az automata a következőt csinálja: az író-olvasó fej valamelyik K szimbólumra lép, és az automata a q állapotban van. Tegyük föl, hogy e K magszimbólumhoz tartozó programrészben i darab vezérlő értékadás van. Először a), b), ill. c) szerint „végrehajtja" az elsőt, ekkor az automata legyen a q_1 állapotban (fent ezt p'-vel jelöltük), utána a második értékadást szimulálja, ekkor a q_2 állapotban van, s ezt folytatja az i-edikig. Ekkor kerüljön a p' állapotba. (A $q, q_1, \ldots, p$ állapotok páronként különbözőek.)

Ez hasonlít egy makróhoz. Amikor egy K szimbólumra lép, egy „*Turing-program*" működése kezdődik el. Ennek befejezése után a következő szimbólumra lép, és egy másik „*Turing-program*" végrehajtása kezdődik.

Az automata működése pedig legyen a következő: A kezdeti, nem-determinisztikus értékadás után (a munkaszalagokon), az író-olvasó fej a bemenőszalagon az első szimbólumra, a munkaszalagokon az utolsó egyes utáni B jelre mutat, és az automata a p_0 kezdőállapotban van (4. ábra).



4. ábra

Ekkor a K_1 szimbólumra kell, hogy mutasson (csak erre értelmezzük, különben hibával leáll):

$(p_0, K_1, B, \ldots) \rightarrow \ldots$ most az író-olvasó fej a K_1 szimbólumra mutat, ekkor a „K_1-makró" végrehajtása kezdődik, azaz a P_1 programrészhez tartozó vezérlő értékadások szimulálása történik, majd:

az első ciklusváltozó szalagja

$$\ldots \rightarrow (p_1, K_1, B, \ldots,; R, H, \ldots),$$

azaz egy p_1 állapotba jutott az automata, és átlépett a következő magszimbólumra, ami K_2 lesz:

$$(p_1, K_2, B, \ldots) \rightarrow \text{„}K_2\text{-makró"} \rightarrow (p_1, K_2, B, \ldots; R, H, \ldots).$$

Figyelnie kell, hogy pontosan annyi K_2 legyen, amennyi a ciklusváltozójának a szalagon számolt értéke. Azaz, ha a szalagon van még egyes, akkor K_2-nek kell megint következnie (ezt biztosítja a DC k utasítás szimulálása), ha pedig nincs több (B jön), akkor K_3 kell, hogy következzen. Ez utóbbi esethez tartozzanak a következő lépések:

$$(p_1, K_3, B, \ldots) \rightarrow (p_2, K_3, B, \ldots; H, L, H, \ldots),$$

és utána

$$(p_2, K_3, B, B, \ldots) \to \text{„}K_3\text{-makró”} \to (p_3, K_3, B, B, \ldots; R, H, \ldots),$$

az első a második ciklusváltozóhoz rendelt szalagon; a második ciklusváltozó szalagján

ugyanakkor a következőre nem tud továbblépni, ami azt jelenti, hogy hibával leáll (nem jut végállapotba):

$(p_2, K_3, 1, \ldots)$ azt jelentené, hogy még nem 0 a ciklusváltozó értéke, de többször nem hajtja végre a ciklust.

A többi hasonló az eddigiekhez, így nem részletezzük. Végül:

$$(p_7, K_7, B, \ldots) \to \text{„}K_7\text{-makró”} \to (p_8, K_7, B, \ldots; R, H, \ldots),$$

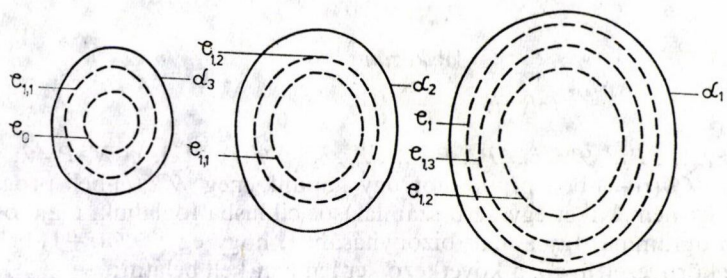
$$(p_8, B, \ldots) \to (p_v, B, \ldots; H, \ldots).$$

p_v legyen az automata végállapota. (Előfordulhat, hogy K_1, K_3, K_5 és K_7 közül valamelyik hiányzik, azaz a P_1, P_3, P_5 és P_7 programrészek közül valamelyik üres. Ekkor a fenti automatát ennek megfelelően módosíthatjuk.)

Az így konstruált *t.l.k. automata* a $K_1K_2^{n_1}K_3K_4^{n_2}K_5K_6^{n_3}K_7$ alakú szavakból (amelyekben n_1, n_2 és n_3 a program szerint összetartozik) álló magnyelvet fogadja el, e nyelv tehát $\mathscr{L}_1$-beli. Igazoltuk tehát, hogy $\mathscr{C}_{1,3} \subseteq \mathscr{L}_1$.

Az előbbi automata konstruálásakor „lényegében” nem használtuk ki, hogy csak 3 ciklus van a programban, azaz tetszőleges $n \in N$ szám esetén készíthetünk ennek mintájára egy olyan *t.l.k. automatát*, amely az n-ciklusú program magnyelvét fogadja el. Ezáltal beláttuk a következő állítást is: Minden $\mathscr{SC}(1)$-beli program magnyelve környezetfüggő.

A kapott összefüggéseket az 5. ábrán szemléltetjük.



5. ábra

Vizsgáljuk tovább a magnyelveket. Vajon milyenek a $\mathscr{C}_2, \mathscr{C}_3, \mathscr{C}_4, \ldots, \mathscr{C}_i, \ldots$ magnyelvosztályok? A következő tétel ad erre a kérdésre választ.

3.1. TÉTEL: Minden egyszerű számlálásos ciklusokból felépíthető URM program magnyelve környezetfüggő (azaz, $\mathscr{C} \subseteq \mathscr{L}_1$).

Bizonyítás: Azt kell belátnunk, hogy egy tetszőleges $\mathscr{SC}(i)$-beli ($i \in N$ tetszőleges) program magnyelve környezetfüggő. i-szerinti teljes indukcióval végezzük a bizonyítást. $i=0,1$-re már láttuk, így tegyük föl: $i \in N$-re igaz, hogy minden $P \in \mathscr{SC}(i)$ magnyelve környezetfüggő. Ebből bizonyítjuk, hogy tetszőleges $\mathscr{SC}(i+1)$-beli program magnyelve is környezetfüggő.

Egy tetszőleges $\mathscr{SC}(i+1)$-beli P program definíció szerint a következő alakú:

P_1
*kezd*1: BR $x1$, *folyt*1
DC $x1$
P_2
JP *kezd*1
*folyt*1: P_3
*kezd*2: BR $x2$, *folyt*2
DC $x2$
P_4
JP *kezd*2
*folyt*2: P_5
$\vdots$
P_{2n-1}
kezdn: BR xn, *folytn*
DC xn
P_{2n}
JP *kezdn*
folytn: P_{2n+1},

ahol $P_2, P_4, \ldots, P_{2n} \in \mathscr{SC}(i)$, $P_1, P_3, \ldots, P_{2n+1} \in \mathscr{SC}(i) \cup \{\lambda\}$, $n \in N$, $P \neq \lambda$. Azaz, egy tetszőleges $\mathscr{SC}(i+1)$-beli programot úgy kapunk meg $\mathscr{SC}(i)$-beli programokból, hogy közülük némelyiket egyszerű számlálásos ciklusba foglaljuk, majd összefűzzük őket egy programmá. Így annak bizonyításához, hogy egy $\mathscr{SC}(i+1)$-beli program magnyelve környezetfüggő, a következő két lemmát kell belátni:

3.2. LEMMA: Ha P egyszerű számlálásos ciklusokból felépíthető program, melynek magnyelve környezetfüggő, és P-t egyszerű számlálásos ciklusba foglaljuk, akkor az így kapott program magnyelve is környezetfüggő lesz.

3.3. LEMMA: Ha $P_1, P_2, \ldots, P_n$ olyan egyszerű számlálásos ciklusokból felépíthető programok, melyeknek magnyelve környezetfüggő, akkor a belőlük összefűzéssel előálló $P = P_1 \ldots P_n$ program magnyelve is környezetfüggő.

Megjegyzés:

(1) $L(P_iP_j)$-ben esetleg szükség van a szimbólumok átírására, hogy azonos programrészekhez azonos, különböző programrészekhez különböző magszimbólumok tartozzanak továbbra is.

(2) Ha a P_i program lineáris programrésszel végződik, a P_j program pedig lineáris programrésszel kezdődik, akkor az összefűzésükkor a két program határán egyetlen lineáris programrész keletkezik, s ennek feleltessünk meg egyetlen magszimbólumot.

A bizonyítást kezdjük a 3.3. lemmával (a 3.2. lemma bizonyításában ugyanis hivatkozunk rá).

A 3.3. lemma bizonyítása: Elég $n=2$-re bizonyítanunk az állítást (ebből ugyanis indukcióval tetszőleges $n>2$-re is következik). Legyen tehát $L(P_1)$ és $L(P_2)$ környezetfüggő, így van olyan A_1 és A_2 *t.l.k. automata,* amelyre $L(A_1)=L(P_1)$, ill. $L(A_2)=L(P_2)$. Tegyük föl, hogy ezek az automaták olyanok, hogy van egy bemenő szalagjuk, és a program minden vezérlő változójához tartozik egy munkaszalag, amin az A_1, ill. az A_2 automata a megfelelő változó értékét számolja. A bemenőszalagon levő szót az automata balról jobbra haladva vizsgálja meg. Az A_1 és A_2 automatáról ezt föltehetjük, mivel a $\mathscr{C}_1$-beli programok magnyelveihez ilyen automatákat készítettünk; és most is ilyen automatát kell készítenünk.

Készítsük el tehát $L(P_1P_2)$-höz az A *t.l.k. automatát* (melynek szintén egy bemenőszalagja és a vezérlő változókhoz egy-egy munkaszalagja van). Először módosítsuk az A_1 és A_2 automatát egy A_1' és A_2' automatává úgy, hogy

— lépéseik közül hagyjuk ki a nemdeterminisztikus kezdeti értékadást,

— vegyünk fel egy-egy munkaszalagot az olyan változók számára is, amelyek sem a P_1, sem a P_2 programban nem vezérlő változók, de a P_1P_2 programban azok (ha vannak ilyenek), és az ezekkel a változókkal végzett műveletek szimulálását is vegyük hozzá az automaták működéséhez.

Az A_1 automata módosítása A_1'-vé:

a) A_1 esetében, ha P_1 ciklussal végződik: ha p_i' állapotban B van a bemenő szalagon (vége a szónak), és a ciklusváltozó szalagján nincs több 1 jel, akkor elfogadja a szót, azaz végállapotba kerül.
A_1' működjön így: p_i' állapotban megvizsgálja, hogy a ciklusváltozó értéke 0-e vagy sem, és ha 0, akkor függetlenül attól, hogy mi van a bemenő szalagon, végállapotba kerül. Ha nem 0, akkor folytatja a működését.

b) A_1 esetén, ha P_1 lineáris programrésszel végződik és P_2 ciklussal kezdődik: az utolsó szimbólum után, ha vége a szónak, elfogadja. A_1'-re teljesüljön a következő: az utolsó szimbólum után bármi jön, kerüljön végállapotba.

c) Ha P_1 lineáris programrésszel fejeződik be, és P_2 lineáris programrésszel kezdődik, akkor az ezen részekhez tartozó K_i és K_j magszimbólumok helyett egyetlen (K_i') magszimbólum lesz. A_1' K_i helyett K_i'-t ismerje föl, és fogadja el a szót anélkül, hogy a „K_i'-makrót" végrehajtaná.

Az A_2 automata módosítása A_2'-vé:

A fenti c) esetben: kezdőállapotában ezentúl a K_i'-t ismerje föl, és hajtsa végre a „K_i'-makrót".

Az így megszerkesztett A_1' és A_2' automatákat a következő módon építjük össze az A automatává:

— Az A automata generáljon nem-determinisztikusan kezdeti értékeket a bemenő vezérlő változók szalagjain (az $\mathscr{SC}(1)$-nél látott módon), majd lépjen vissza a bemenőszalagon az első szimbólumra, q_0 állapotban.
— q_0 egyezzék meg az A_1' automata kezdőállapotával, ezáltal az A működése az A_1' működésével kezdődik. Az A_1' működése akkor fejeződik be, amikor az író-olvasó fej a P_1-hez tartozó szó utolsó szimbóluma utáni szimbólumra mutat, és az A_1' automata végállapotba jutott.
— Az A_1' végállapota egyezzék meg az A_2' kezdőállapotával. Ekkor az A_2' automata kezdi meg működését, és az utolsó szimbólum utáni B jelen fog megállni, q_v állapotban (hacsak közben hiba nem történt), ami legyen az A automata végállapota.

Ekkor az A automatán belül az A_1' és A_2' automata a munkaszalagokon számolja a ciklusváltozók értékét, úgy, hogy amikor az A_2' automata működésbe lép, a „P_2 programrész" minden ciklusváltozójának kezdeti értékét megkapja a munkaszalagokon.

Vagyis az így konstruált *A t.l.k. automata* (amelyről a 3.1. lemmából tudjuk, hogy a munkaszalagjaiból használt részek hossza felülről becsülhető a bemenő szó hosszának konstansszorosával) a P_1P_2 program magnyelvét fogadja el, és úgy működik, hogy balról jobbra halad a bemenőszalagon, miközben a munkaszalagokon a változók értékeit számolja.

Ezzel a 3.3. lemma bizonyítását befejeztük.

Megjegyzés: Ebből a lemmából is következik, hogy (minden $n \in N$-re) az $\mathscr{SC}(1)$-beli n-ciklusú programok magnyelvei környezetfüggőek, így ennek bizonyításához nincs szükség automata készítésére. Az 1-, 2- és 3-ciklusú programok magnyelveiről ugyanis már láttuk, hogy környezetfüggőek, és (tetszőleges $n \geqq 4$-re) az n-ciklusú programokat ezek összefűzésével kapjuk meg.

A 3.2. lemma bizonyítása: A P egyszerű számlálásos ciklusokból felépített program, melynek magnyelve, $L(P)$, környezetfüggő. P-t egyszerű számlálásos ciklusba foglaljuk (x nem változója P-nek):

kezd: BR x, *folyt*

DC x

P

JP *kezd*

Ezt a programot P'-vel jelöljük, s azt állítjuk, hogy $L(P')$ környezetfüggő. Tegyük föl, a $(\mathrm{DC}\ x)P$ programot P_1-gyel jelölve, hogy a 3.3. lemma szerint az $L(P_1)$-hez készített *t.l.k. automata, A* olyan, hogy van egy bemenőszalagja — ezen a bemenő szó van —, amin az író-olvasó fej balról jobbra halad, és minden vezérlő változó értékét egy munkaszalagon számolja.

Módosítsuk az A automatát (lényegében „visszacsatolás" beiktatásával), így:
— Lépései közül hagyjuk ki a kezdeti értékadást, és vegyünk hozzá egyrészt egy-egy munkaszalagot az x-nek és az olyan P-beli változóknak a szimulálására, amelyek P_1 egyszeri végrehajtásakor nem vezérlő változók, de ha többször hajtjuk végre P_1-et, akkor azok lesznek; másrészt a most említett változók értékeinek az új

szalagokon való számolását (az x változót, mint ciklusváltozót kezelje, a DC utasítás szimulálásával).

— Az A automata végállapotba kerülését változtassuk meg a 3.3. lemma bizonyításában szereplő a) és b) pont szerint.

Az így kapott automatát jelöljük A'-vel. Az $L(P')$-t elfogadó A_1 automata működése pedig legyen a következő:

— generáljon nem-determinisztikusan kezdőértéket a bemenő vezérlő változók munkaszalagjain, majd kerüljön az első szimbólumra (ha van) mutatva, az A' automata p_v végállapotába.

— p_v állapotban, ha az automata a bemenőszalagon nem B jelre mutat, kerüljön az A' automata p_k kezdőállapotába, ekkor az A' automata működése megkezdődik.

— Ha p_v állapotban a bemenőszalagon B jelet olvas, vizsgálja meg, hogy az x ciklusváltozó szalagján 0 van-e, és ha igen, akkor fogadja el a szót, azaz kerüljön a q_v állapotba, ami az A_1 automata végállapota lesz.

Ezzel olyan *t.l.k. automatát* konstruáltunk, amely ezt a nyelvet fogadja el (a munkaszalagok hossza ugyanis, a 3.1. lemmából tudjuk, felülről becsülhető a bemenő szó hosszának konstans-szorosával). Így $L(P')$ környezetfüggő.

Ezzel a tétel bizonyítását is befejeztük.

Megjegyzés: A bizonyításban megadott automata is olyan, hogy egy bemenőszalagja van — amin balról jobbra egyszer halad végig —, és munkaszalagokon számolja a változók értékeit. Így mindkét lemma bizonyításában valóban föltehettük, hogy ilyen automatáink vannak.

4. A tartalmazási viszony pontosítása a vizsgált magnyelvosztály, valamint a környezetfüggő és a környezetfüggetlen nyelvosztály között

A 3. fejezetben beláttuk, hogy minden egyszerű számlálásos ciklusokból felépíthető URM-program magnyelve környezetfüggő (azaz, $\mathscr{C} \subseteq \mathscr{L}_1$). Vajon ez fordítva is igaz-e, azaz minden környezetfüggő nyelvhez megadható-e egy olyan egyszerű számlálásos ciklusokból felépített program, aminek e nyelv magnyelve?

Ebben a fejezetben bebizonyítjuk, hogy $\mathscr{C}$ valódi része $\mathscr{L}_1$-nek, és — ezzel összefüggésben — hogy $\mathscr{C}$ tartalmazás szempontjából összehasonlíthatatlan $\mathscr{L}_3$-mal, $\hat{\mathscr{L}}_2$-vel és $\hat{\mathscr{L}}_1$-gyel (azt már láttuk, hogy $\mathscr{C} \cap \mathscr{L}_3 \neq \emptyset$, $\mathscr{C} \cap \hat{\mathscr{L}}_2 \neq \emptyset$ és $\mathscr{C} \cap \hat{\mathscr{L}}_1 \neq \emptyset$). Ezek az állítások abból fognak következni, hogy — amint azt meg fogjuk mutatni — bizonyos $\mathscr{C}$-beli nyelvek kommutatív lezárásai $\mathscr{C}$-n kívül esnek. (Ebben a fejezetben nem ragaszkodunk ahhoz, hogy egy magnyelv mindig valamilyen $\{K_1, \ldots, K_n\}$ alakú ábécé fölötti nyelv legyen, s a $\mathscr{C}$ osztályt is ennek megfelelően, formailag általánosabban értelmezzük.)

Legyen $m \in N_+$ és $a_1, \ldots, a_{m+1}, b_1, \ldots, b_m$ $2m+1$ különböző magszimbólum, és legyen $L_m := \{a_1 b_1^n a_2 b_2^n \ldots a_m b_m^n a_{m+1} \mid n \in N_+\}$. Világos, hogy $L_m \in \mathscr{C}_{1,m}$ (pl. $b_1, \ldots, b_m$ m független ciklusnak felel meg, az a_1-nek megfelelő programrész pedig az m db ciklushoz tartozó ciklusváltozókat azonos kezdeti értékre állítja be). És azt is tudjuk (pl. a 3. fejezetben is láttuk), hogy $L_1 \in \mathscr{L}_3$, $L_2 \in \hat{\mathscr{L}}_2$, és $m \geqq 3$-ra $L_m \in \hat{\mathscr{L}}_1$.

Most vizsgáljuk meg ezen nyelvek kommutatív lezárásait. A következő állításokat fogjuk belátni róluk.

4.1. LEMMA: $m \in N_+$-ra, $\mathrm{com}(L_m) \notin \mathscr{C}$.

4.2. LEMMA: Legyen L tetszőleges környezetfüggetlen nyelv, T az L szavaiban előforduló jelek (betűk) halmaza. Vegyünk $x_1, \ldots, x_s$ ($s \in N_+$) véges sok jelet úgy, hogy $x_i \notin T$ ($1 \leqq i \leqq s$), és ezek mindegyikét pontosan egyszer helyezzük el L szavaiban bárhol. Az így kapott L' nyelv is környezetfüggetlen.

4.3. LEMMA: $\mathrm{com}(L_1) \in \mathscr{L}_3$, $\mathrm{com}(L_2) \in \hat{\mathscr{L}}_2$.

4.4. LEMMA: Ha $L \in \mathscr{L}_1$, és L rendezett, akkor $\mathrm{com}(L) \in \mathscr{L}_1$.

Megjegyzés: Ez utóbbi állítás következik egy általánosabb tételből (SZIJÁRTÓ [13, Theorem 4.1]), amely szerint $\mathscr{L}_0$ és $\mathscr{L}_1$ zárt tetszőleges „függetlenségi reláció" szerinti lezárására, nekünk azonban csak erre a gyengébb állításra lesz szükségünk, és az alábbiakban közvetlen bizonyítást adunk rá.

A 4.1. lemma bizonyítása: Elég két $\mathrm{com}(L_m)$-beli szóról belátni, hogy nem tartozhatnak ugyanannak a programnak a magnyelvébe. Legyen a két szó:

$$w_1 := a_1 b_1^n a_2 b_2^n \ldots a_m b_m^n a_{m+1}$$

és

$$w_2 := a_{m+1} b_1^n a_2 b_2^n \ldots a_m b_m^n a_1.$$

Tegyük föl, hogy létezik olyan P egyszerű számlálásos ciklusos URM-program, amelynek $L(P)$ magnyelvében w_1 is és w_2 is benne van.

Tekintsük P-nek azt a $P_1 (\in \mathscr{PC}(0))$ részét, amely a P w_1 szerinti W_1 lefutása során a w_1 elején álló a_1-nek felel meg. (Elképzelhető, hogy P_1-nek további előfordulása is van P-ben, s akkor szükségszerűen valamilyen ciklusban, illetve ciklusokban, de P_1-gyel az előbbi konkrét előfordulást fogjuk jelölni.)

Tegyük fel, hogy P_1 valamilyen C ciklusban van, s legyen C ciklusváltozója x. Ekkor P_1 ciklusba zártsága csak egyszeres mélységű lehet, különben a W_1 lefutásban P_1 előtt legalább egy DC utasítás is végrehajtódna, vagyis w_1-ben a szókezdő a_1 előtt még legalább egy másik jelnek is kellene állnia. Másrészt, ha P-nek a C ciklus előtti része nem üres, akkor csupa olyan ciklusokból kell állnia, amelyek a W_1 lefutásban üresen futnak le, ezért ciklusváltozóik értéke W_1 kezdetén 0, x-é pedig 1 (így x a P-nek bemenő változója). Ha tehát P-t a változóinak olyan kezdő értékrendszerével indítjuk el, amely a W_1-hez tartozó kezdő értékrendszertől csak abban különbözik, hogy x értéke nagyobb 1-nél, akkor P_1 is legalább x-szer futna le, s $L(P)$-nek egy olyan szavát kapnánk, amelyben a_1 is legalább $x(>1)$-szer fordulna elő, ami lehetetlen. Ezért P_1 a P-ben szabadon (nem ciklusban) áll.

Ugyanezt a gondolatmenetet alkalmazva w_1 és a_1 helyett w_2-re és a_{m+1}-re, azt kapjuk, hogy a P w_2 szerinti lefutásában a w_2 elején álló a_{m+1}-nek megfelelő $P_{m+1} (\in \mathscr{PC}(0))$ programrész is szabadon áll P-ben (és korábbi megállapodásunk értelmében, természetesen, $a_1 \neq a_{m+1}$ miatt $P_1 \neq P_{m+1}$). Ez viszont azt jelenti, hogy a_1 és a_{m+1} egymáshoz képesti sorrendje az $L(P)$ minden szavában ugyanaz, mint P_1 és P_{m+1} egymáshoz képesti sorrendje P-ben, ellentmondásban azzal, hogy w_1 is és w_2 is $L(P)$-hez tartozik. A $\mathrm{com}(L_m) = L(P)$ egyenlőség tehát nem teljesül semmilyen egyszerű számlálásos ciklusos P URM-programra. Ezzel a 4.1. lemmát bebizonyítottuk.

A 4.2. lemma bizonyítása: Az állítást elég az $s=1$ esetre belátni (ebből ugyanis indukcióval tetszőleges $s\in N_+$-ra is következik). L környezetfüggetlen, így van olyan A veremautomata, ami L-et — végállapotba kerülve — elfogadja. Legyen $A=(Z, K, T, M, z_0, q_0, H)$, ahol

$T=\{a_1, \ldots, a_n\}$, $n\in N$

$K=\{q_i|$, ahol $0\leqq i\leqq k\}$, $k\in N$ rögzített, és az M átmenetfüggvény a következő alakú szabályokból áll:

$$M(z, q, a) = \{(w_1, q_1), \ldots, (w_m, q_m)\}.$$

$x\notin T$, ezt a betűt bárhol elhelyezhetjük L szavaiban. Az így kapott L'-höz készítsünk egy veremautomatát A segítségével:

$$\acute{A} = \{Z, \bar{K}, T\cup\{x\}, \bar{M}, z_0, q_0, \bar{H}\},$$

ahol

$$K := \{q_i, \bar{q}_i|0 \leqq i \leqq k\}.$$

$\bar{H}$ legyen a következő:

$$\bar{q}_i\in\bar{H} \Leftrightarrow q_i\in H,$$

és az $\bar{M}$ átmenetfüggvényt definiáljuk a következőképpen:

$$\bar{M}(z, q, a) = M(z, q, a),$$

$$\bar{M}(z, q, x) = \{(z, \bar{q})\},$$

és

$$\bar{M}(z, \bar{q}, a) = \{(w_1, \bar{q}_1), \ldots, (w_m, \bar{q}_m)\},$$

ha

$$M(z, q, a) = \{(w_1, q_1), \ldots, (w_m, q_m)\}.$$

Az így kapott veremautomata addig, amíg x-et meg nem találta a bemenő szóban, ugyanúgy működik, mint A, x felismerésekor a q állapotból átmegy a neki megfelelő $\bar{q}$ állapotba, s ettől kezdve továbbra is ugyanúgy működik, mint A, csak p állapot helyett $\bar{p}$ állapotban. Egy szót akkor fogad el, ha megtalálta benne x-et (pontosan egyszer), és x-et kihagyva belőle, A elfogadná a szót. Vagyis valóban L'-t fogadja el, ezzel a bizonyítást befejeztük.

A 4.3. lemma bizonyítása: Mivel a $\mathrm{com}\,(L_1)\in\mathscr{L}_3$ nyilvánvaló, csak a $\mathrm{com}\,(L_2)\in$ $\in\hat{\mathscr{L}}_2$ bizonyításával foglalkozunk. Azt kell belátni, hogy

$$\text{I. } \mathrm{com}\,(L_2)\in\mathscr{L}_2, \quad \text{de} \quad \text{II. } \mathrm{com}\,(L_2)\notin\mathscr{L}_3.$$

I. Az $L=\{b^n d^n|\, n\geqq 1\}$ nyelv kommutatív lezárásáról, a $\mathrm{com}\,(L)=\{w||w|_b=$ $=|w|_d\}$ nyelvről ismeretes, hogy környezetfüggetlen (például [14]-ből). Alkalmazzuk erre a 4.2. lemmát: $\mathrm{com}\,(L)$ minden szavába véges sok (pontosan három darab: a, c, e) betűt helyezve, a $\mathrm{com}\,(L_2)$ nyelvet kapjuk, ami a 4.2. lemma szerint környezetfüggetlen.

II. Vegyük a $\mathrm{com}\,(L_2)$ metszetét az $a_1 b_1^+ a_2 b_2^+ a_3$ nyelvvel:

$$a_1 b_1^+ a_2 b_2^+ a_3 \cap \mathrm{com}\,(L_2) = L_2.$$

$a_1 b_1^+ a_2 b_2^+ a_3$ reguláris nyelv. Alkalmazzuk a 2.6. tételt. Ha com $(L_2) \in \mathscr{L}_3$ lenne, akkor $L_2 \in \mathscr{L}_3$ is teljesülne, de ez nyilvánvaló ellentmondás, hiszen tudjuk, hogy $L_2 \in \hat{\mathscr{L}}_2$. Így com $(L_2) \notin \mathscr{L}_3$.

Ezzel a 4.3. lemma bizonyítását befejeztük, vagyis com $(L_2) \in \hat{\mathscr{L}}_2$.

Megjegyzés: A 4.1. és 4.3. lemmából következik, hogy com $(L_1) \in \mathscr{L}_3 \setminus \mathscr{C}$, és com $(L_2) \in \hat{\mathscr{L}}_2 \setminus \mathscr{C}$.

A 4.4. lemma bizonyítása: Mivel L környezetfüggő, létezik olyan lineárisan korlátos automata (A), ami elfogadja. Ezért com (L)-hez olyan A' lineárisan korlátos automatát kell készíteni, amely com (L) szavait rendezett alakra hozza, és végállapota megegyezik A kezdőállapotával. A két automata együtt pontosan com (L)-et elfogadó többszalagú lineárisan korlátos automata lesz, s ezzel bizonyítjuk, hogy com (L) környezetfüggő (lásd a 2.2. tételt).

Tegyük föl, hogy $L \in \{a_1, \ldots, a_k\}^*$. A'-t definiáljuk a következőképpen:

— 2 szalagja van, az elsőn a bemenő szöveg, a második kezdetben üres;

— p_0 legyen a kezdőállapota, $p_1, \ldots, p_{k-1}$ állapotai, p_k pedig a végállapota. Közülük $p_0, \ldots, p_{k-1}$ A-nak nem állapotai, p_k pedig A kezdőállapota;

— működése: ha p_i állapotban $(0 \leqq i \leqq k-1)$ az első szalagon

a) a_{i+1}-et olvas, akkor írjon a_{i+1}-et a második szalagra és egyet lépjen jobbra, az első szalagon pedig, ha i páros, jobbra, ha i páratlan, balra lépjen egyet;

b) a_j-t $(j \neq i+1)$ olvas, akkor az első szalagon lépjen egyet jobbra vagy balra ugyancsak i-től függően;

c) B-t olvas, kerüljön p_{i+1} állapotba, és az első szalagon lépjen egyet visszafelé; végül a p_k állapotba kerül, ami A kezdőállapota.

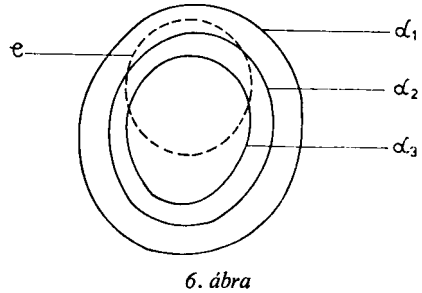
Mindkét szalagon csak annyi helyet használtunk fel, mint amilyen hosszú a bemenő szó. Így többszalagú lineárisan korlátos automatát konstruáltunk, s ezzel a 4.4. lemma bizonyítását befejeztük.

Megjegyzés: A 4.1. és 4.4. lemmából következik, hogy $\mathscr{C}$ valódi része $\mathscr{L}_1$-nek $(\mathscr{C} \subset \mathscr{L}_1)$ hiszen $L_m\,(m \geqq 3)$ rendezett, így com $(L_m) \in \hat{\mathscr{L}}_1 \setminus \mathscr{C}$.

A lemmákból következik az alábbi tétel.

4.1. TÉTEL: $\mathscr{L}_3 \setminus \mathscr{C} \neq \emptyset$, $\hat{\mathscr{L}}_2 \setminus \mathscr{C} \neq \emptyset$, és $\hat{\mathscr{L}}_1 \setminus \mathscr{C} \neq \emptyset$.

Ez a tétel pontosítja a $\mathscr{C}$, $\mathscr{L}_1$, $\mathscr{L}_2$ és $\mathscr{L}_3$ közötti eddig megismert kapcsolatokat, ezt a 6. ábrán szemléltetjük.



6. ábra

Köszönetnyilvánítás: Befejezésül szeretném köszönetemet kifejezni HORVÁTH SÁNDORnak, aki munkám során végig értékes szakmai segítséget nyújtott. Köszönetet mondok még VARGA LÁSZLÓnak a cikk végső formájának kialakításával kapcsolatos hasznos tanácsaiért, és KOMOR TAMÁSnak a dolgozat gondos átnézéséért és értékes megjegyzéseiért.

IRODALOM

[1] BERECZKI, I., „Nem-elemi rekurzív függvény létezése", *Az Első Magyar Matematikai Kongresszus Közleményei* (Comptes Rendus du Premier Congrès des Mathématiciens Hongrois), Budapest, 1950, 409—417.
[2] BRAINERD, W. S., LANDWEBER, L. H., *Theory of Computation* (John Wiley and Sons, New York, London, 1974).
[3] DÖRNYEI, Á., „Magnyelvek egy osztályának formális nyelvi vizsgálata", Diákköri dolgozat, ELTE, TTK, Numerikus és Gépi Matematikai Tanszék, Budapest, 1980.
[4] ENGELER, E., *Introduction to the Theory of Computation* (Academic Press, New York, London, 1973).
[5] GRZEGORCZYK, A., "Some classes of recursive functions", *Rozprawy Matematyczne* **4**.
[6] HARRISON, M. A., *Introduction to Formal Language Theory* (Addison-Wesley, Reading, Mass., 1978).
[7] KALMÁR, L., „Egyszerű példa eldönthetetlen aritmetikai problémára", *Matematikai és Fizikai Lapok* **50** (1943) 1—23.
[8] MEYER, A. R., RITCHIE, D. M., "The complexity of loop programs", *Proceedings of the ACM National Meeting*, 1967, 465–469.
[9] PRATT, T., "Program analysis and optimization through kernel-control decomposition" *Acta Informatica* **9** (1978) 195–216.
[10] RÉVÉSZ, GY., *Bevezetés a formális nyelvek elméletébe* (Akadémiai Kiadó, Budapest, 1979).
[11] SCHNORR, C. P., *Rekursive Funktionen und ihre Komplexität* (B. G. Teubner, Stuttgart, 1974).
[12] SHEPHERDSON, J. C., STURGIS, H. E., "Computability of recursive functions", *Journal of the ACM* **10** (1963) 217–255.
[13] SZÍJÁRTÓ, M., "Trace languages and closure operations", *Automataelméleti Füzetek*, az ELTE TTK Numerikus és Gépi Matematikai Tanszékének kiadványa, 1979/2. sz.
[14] VARGA, L., *Rendszerprogramok elmélete és gyakorlata* (Akadémiai Kiadó, Budapest, 1978).

(Beérkezett: 1981. január 5.)

DÖRNYEI ÁGNES
MTA KÖZPONTI FIZIKAI KUTATÓ INTÉZET
1525 BUDAPEST XII., KONKOLY THEGE ÚT 29—33.

ON THE STRUCTURE OF KERNEL LANGUAGES OF SIMPLE COUNTING-LOOP PROGRAMS

Á. DÖRNYEI

PRATT [9] defines the decomposition of an arbitrary (graph-like) program into "control part" and "kernel part", and on the basis of this decomposition, the "kernel language" of the program, as the set of finite symbolic execution sequences of the program. The kernel languages have a connection with programming, because they well reflect the behaviour of programs, and therefore they are related to program proving, to program testing by symbolic execution, and to program optimization. The aim of the present paper is to begin a general study of the structure of kernel languages. We slightly modify PRATT's original definition, in order that each nonempty word in a kernel language represents a whole computation of the program in question. We prove that the class of kernel languages of simple counting-loop URM programs is a proper subclass of the class of context-sensitive languages, and (in terms of set-theoretical inclusion) it is incomparable with the classes of strictly context-sensitive, strictly context-free and regular languages. The main tools (taken from formal language theory) used in the proofs are: many-tape linear bounded automata, pushdown automata, and the commutative closure of languages.

Alkalmazott Matematikai Lapok 7 (1981) 311—329

KONKURRENCIAVEZÉRLÉS ELOSZTOTT ADATBÁZISOKON; EGY IDŐZÍTÉSEN ALAPULÓ ALGORITMUS

POGÁNY ANDRÁS

Budapest

Dolgozatunkban a centrális rendszerek kapcsán ismertetjük a szinkronizáció alapfogalmait, valamint néhány algoritmusát. Az elosztott rendszerek közül az elosztott adatbázisokkal foglalkozunk. Definiáljuk a tématerületen használatos legfontosabb fogalmakat. A konkurrencia vezérlő algoritmusok két nagy csoportját különböztetjük meg: az adatbázis teljes lezárása nélkül, illetve az adatbázis teljes lezárásával dolgozó algoritmusokat. Röviden tárgyaljuk az egyes csoportok jellegzetességeit, majd ismertetünk két algoritmust. Végül megadjuk a leírását egy új konkurrencia vezérlési algoritmusnak, amely az előzőektől a műveletek megerősítési fázisának megvalósításában különbözik. Az eljárás, a csomópontok értesítését egy szinkronizáció sikerességéről (megerősítés) nem üzenetekkel, hanem a csomópontokban indított időzítésekkel oldja meg. Így egy művelet minimális számú üzenettel hajtható végre. Ismertetjük az algoritmus megvalósításakor felmerülő problémákat, és javaslatot teszünk azok megoldására.

1. Bevezetés

A konkurrens vezérlések problémáját nem az elosztott adatbázisok vetették fel. A téma messzemenően nem újkeletű, a számítástechnikát tekintve szinte ősi. Valamikor a 60-as években a multiprogramozás kialakulásával merült fel először. Abban az időben természetesen még nem léteztek elosztott rendszerek (ehhez előbb a számítógéphálózatoknak kellett kialakulniuk), de a probléma lényege centrális rendszerekben is ugyanaz.

Dolgozatunk célja egy új algoritmus ismertetése az elosztott adatbázisok konkurrens felújításainak vezérlésére. Ahhoz, hogy algoritmusunk érthető és értékelhető legyen, ismertetjük a tématerületen használatos alapfogalmakat és definíciókat, a megoldandó problémát, valamint néhány hasonló célra használt algoritmust.

Mellőztük az ismertetett algoritmusok helyességének bizonyításait; ehelyett inkább a minél szélesebb körű ismertetésre törekedtünk. Az algoritmusok pontos leírását az érdeklődő olvasó a hivatkozott irodalmakban megtalálhatja.

Az első fejezet a centrális rendszerekkel foglalkozik. Az itt megadott alapfogalmakat és definíciókat a későbbiekben is felhasználjuk. A második fejezetet az elosztott rendszereknek szenteltük. Ezen belül — az algoritmusunkhoz kapcsolódóan — azoknak egy fontos válfaját, az elosztott adatbázisokat tárgyaljuk. Az itt megadott algoritmusok osztályozása kissé önkényes; azt a célt szolgálja, hogy részletesebben ismertethessünk két algoritmust, amelyek az általunk javasolthoz közel állnak. A harmadik fejezetben saját algoritmusunkat ismertetjük. Ennek újszerűsége az időzítés bevezetésében rejlik, amivel minimális művelet-vérehajtási idő érhető el. Bár az eljárás még nem minden részletében kidolgozott, igyekeztünk minden, az alkalmazás során (esetleg) felmerülő problémát előre megvizsgálni.

2. Szinkronizáció centrális rendszerekben

Alapfogalmak; definíciók

A folyamatok és erőforrások a számítástechnikai rendszerekben alapfogalmak. Érzékeltetésükhöz megadjuk körülírásukat, amelyet egy példával illusztrálunk.

Folyamat: a rendszeren belül egyértelműen azonosítható algoritmus, (illetve algoritmusok egymásutánisága), amely(ek)nek előrehaladásukhoz szükségük van a rendszer erőforrásaira.

Erőforrás: a rendszer bármely olyan eszköze, amely a folyamatok rendelkezésére áll ahhoz, hogy azok feladatukat megoldhassák.

A jelen dolgozat szempontjából minden számítástechnikai rendszer folyamatok és erőforrások halmazából álló együttesnek tekinthető.

Példa: Legyen rendszerünk egy város. A rendszer *erőforrásai* a házak, utak, közlekedési eszközök, közlekedési szabályok, emberek, közlekedést szabályozó berendezések, a város térképe stb. A rendszerben *folyamatnak* definiálhatunk *egy utazást* a város A pontjából B pontjába. Ha egy ilyen utazás megvalósul, akkor egy meghatározott algoritmus szerint valósul meg. Történetesen: autóval, a közlekedési szabályokat figyelembe véve, az utak használatával, a megfelelő útvonal kiválasztásával. Ezen algoritmust mégsem azonosíthatjuk magával a folyamattal, mert a folyamat akkor is egy utazás lett volna A-ból B-be, ha történetesen villamossal és autóbusszal történik. Az az algoritmus tehát, amely szerint a folyamatok előrehaladnak, ugyancsak a rendszer része, a rendszer erőforrása. Erőforrás például a közlekedési eszköz megválasztásának algoritmusa.

2.1. DEFINÍCIÓ. *Oszthatatlan erőforrás:* olyan erőforrás, amelyet egy időben csak egyetlen folyamat vehet igénybe (birtokolhat).

2.2. DEFINÍCIÓ. *Megosztható erőforrás:* olyan erőforrás, amelyet egy időben több folyamat is birtokolhat.

Egy erőforrás a folyamatok bizonyos halmazára nézve oszthatatlan, vagy megosztható. Tehát az erőforrásokat bizonyos folyamatok csak kizárólagosan, mások megosztva használhatják.

Példa: Ha egy autót többen akarnak igénybe venni, akkor az egy irányban haladók megosztva használhatják, de egy kerékpárt egy időben csak egyetlen személy vehet igénybe.

A folyamatoknak a rendszeren belül önálló életük van, egymással párhuzamosan haladnak előre. A folyamatok akkor találkoznak, ha a rendszer valamely erőforrására egyidőben tartanak igényt.

2.3. DEFINÍCIÓ: *Folyamatok szinkronizálása:* (folyamatok erőforrás hozzáférésének szinkronizálása) olyan algoritmus, amely biztosítja, hogy egy folyamat egy erőforrást akkor birtokoljon, ha

a) — az erőforrás a folyamatra nézve kizárólagos használatú — más folyamat nem birtokolja;

b) — az erőforrás a folyamatra nézve megosztott használatú — az erőforrást egyetlen folyamat sem birtokolja kizárólagosan, és megosztva kevesebben birtokolják, mint az erőforrásra megengedett maximum.

Szinkronizációs algoritmusok

Amíg egy rendszerben egyidejűleg csak egyetlen folyamat él, vagy a párhuzamos folyamatok erőforrás-igényei diszjunktak, addig a konkurrens erőforrás-használat problémája nem merül fel. A szinkronizáció akkor válik szükségessé, ha a rendszerben párhuzamos folyamatok vannak, amelyek közül egyidőben többen is igényelhetnek egy adott erőforrást.

Centrális rendszerekre (egyetlen számítógép, egyetlen közös memória) DIJKSTRA [3] javasolt egy megoldást, amely azóta klasszikussá vált, s bizonyos, itt megfogalmazott elveket még az elosztott algoritmusokban is felhasználnak.

2.4. DEFINÍCIÓ. *Erőforrás szemafor:* Egész értékű változó, amelynek kezdőértéke megegyezik az erőforrást egyidőben igénybe vehető folyamatok számával.

Egy erőforrás szemaforjának kezdőértéke 1, ha az erőforrás kizárólagos használatú.

2.5. DEFINÍCIÓ. *Oszthatatlan művelet:* Olyan művelet, amelyet egy időben csak egyetlen folyamat hajthat végre. Ha egy ilyen művelet végrehajtásába egy folyamat belekezd, akkor semmi sem akadályozhatja meg, hogy azt befejezze.

2.6. DEFINÍCIÓ. *P művelet:* Oszthatatlan műveletként a szemafor értékét eggyel csökkenti, és

a) ha a szemafor értéke $(SZÉ) \geqq \varnothing$, a folyamatnak birtokba adja az erőforrást;

b) ha $SZÉ < \varnothing$ a folyamatot várakoztatja.

2.7. DEFINÍCIÓ. *V művelet:* Oszthatatlan műveletként a szemafor értékét növeli eggyel, és

a) ha $SZÉ \geqq \varnothing$, akkor a várakozók közül az elsőt tovább engedi (az erőforrást a birtokába adja);

b) ha $SZÉ > \varnothing$ nem tesz semmit.

A *P* és *V* műveletek csak a szemafor állításában oszthatatlanok.

2.8. DEFINÍCIÓ. *Várakozás:* Egy folyamat várakoztatása azt jelenti, hogy megakadályozzák annak előrehaladását, nem engedjük, hogy további műveleteket hajtson végre.

2.9. DEFINÍCIÓ. *Holtpont.* Egy rendszerben holtpont van, ha benne folyamatoknak olyan $F1, \ldots, Fn$ sorozata keletkezik, amelyben minden Fi előrehaladásához szükséges erőforrás $Fi+1$ birtokában van; végül Fn erőforrását $F1$ birtokolja.

(Ilyen folyamatok külső beavatkozás nélkül sohasem tudják befejezni feladatukat).

A *P* és *V* műveleteket szokás szinkronizációs primitíveknek nevezni. A szinkronizáció ezen primitívek és a szemafórok segítségével a következőképpen valósítható meg: Minden erőforráshoz rendeljünk egy megfelelő kezdőértékű szemafort. A folyamatok, ha erőforrást igényelnek, hajtsanak végre egy *P*-műveletet az erőforráshoz rendelt szemafóron. Mikor a *P*-művelet befejeződik, az erőforrás a folyamat birtokában van (a folyamat közben várakozhatott, de a várakozás a folyamat szempontjából a *P*-művelethez tartozik). A folyamat az erőforrás birtoklását egy *V* művelet végrehajtásával fejezi be.

Ha a folyamatok minden megkötöttség nélkül igényelhetnek egyre újabb és újabb erőforrásokat, akkor a fenti eljárás holtpont-veszélyes. A megkötöttségeket,

amelyeket a folyamatoknak az erőforrás igénylésben be kell tartaniuk, ahhoz hogy a rendszer holtpontmentes legyen, HABERMANN [4] vizsgálta.

Az algoritmusnak van egy gyenge pontja; a rendszerben akkor is holtpont keletkezhet, ha minden folyamat betartja a holtpontmentes erőforrásigénylés szabályait. Ugyanis a folyamatoknak aktívan kell cselekedniük ahhoz (V művelet végrehajtása), hogy egy erőforrás birtoklását megszüntethessék. Ez akár programhiba folytán, akár azért, mert a folyamat az erőforrás birtoklása közben valamilyen oknál fogva (géphiba, illegális művelet) felszámolódott, elmaradhat, s ezzel az erőforrás más folyamatok számára hozzáférhetetlenné válik.

A problémára LAMPORT [5] hívta fel a figyelmet, s mindjárt megoldást is javasolt, amely az irodalomban „*bakery*" algoritmus néven ismert. Az eljárás azon az elven alapszik, amelyet nálunk az orvosi rendelőkben, Amerikában pedig a pékségekben alkalmaznak. A helyiségbe lépve mindenki szakít egy sorszámot, s az emberek helyett a sorszámok állnak sorban. A várakozó akkor kerül sorra, ha sorszáma a legkisebbé válik.

Az algoritmus megvalósítása: minden folyamatnak minden erőforráshoz van egy számlálója, SZ_{ik} (az i-edik folyamat k-adik erőforrás-számlálója). A folyamatok egymás számlálóihoz hozzáférhetnek. Ha az i-edik folyamat a k-adik erőforrást birtokba akarja venni, akkor a következőképpen jár el:

$$SZ_{ik} = \max \{SZ_{jk}: j = 1, n\} + 1;$$

azaz megvizsgálja az összes folyamat ezen erőforráshoz rendelt számlálóját, s a saját számlálóját ennél eggyel nagyobb értékre állítja (letépi a következő sorszámot). Majd mindaddig várakozik, amíg a rendszerben nála kisebb sorszám van.

várakozás=igaz
DO WHILE várakozás=igaz
várakozás=hamis
DO $j=1, n$
IF $SZ_{ik} > SZ_{jk}$ THEN várakozás=igaz
END
END
műveletek az erőforráson
$SZ_{ik}=\emptyset$

Ha egy folyamatnak nincs többé szüksége az erőforrásra, a hozzá tartozó számlálóját nullázza.

A *Dijkstra algoritmussal* ellentétben a folyamatoknak itt nem szükséges aktívan részt venniük az erőforrás felszabadításában, nem kell műveletet végrehajtaniuk, hanem csak egy változójuknak kell nulla értéket felvennie. Ez utóbbi könnyen elérhető, ha a folyamatokkal együtt erőforrás-számlálóikat is megsemmisítik.

A „*bakery*" algoritmus már tartalmazza egyfajta elosztott szinkronizáció magját. Ugyanis az előbbi algoritmussal szemben (ahol minden erőforráshoz egyetlen központi szemafor tartozik és a folyamatok ezen hajtanak végre műveleteket), a folyamatok mindegyike a saját adatterületén dolgozik, a többi folyamatról csak

információkat szerez. Az eljárás könnyedén elosztottá tehető, ha az egyik folyamat a másik folyamat adatmezejének olvasása helyett (erőforrás számláló) azzal üzenetet vált, és így értesül a számláló aktuális értékéről.

Ezt a kiterjesztést hajtotta végre RICART és AGRAWALA [7].

3. Szinkronizáció elosztott adatbázisokon

Alapfogalmak; definíciók

3.1. DEFINÍCIÓ. *Elosztott rendszerek:* A számítási folyamatokat nem egyetlen számítógép, hanem több egymással összekötött számítógép oldja meg. A feladatokat általában részfeladatokká bontják, és ezeket az egyes számítógépek (csomópontok) külön-külön egymással párhuzamosan oldják meg, majd az eredményt összegyűjtik.

Az elosztott rendszerek mindig feltételeznek egy számítógép-hálózatot, amely biztosítja az egyes csomópontok egymással való információ cseréjét.

3.2. DEFINÍCIÓ. *Elosztott adatbázis:* Olyan elosztott rendszer, amelynek a számítási feladatai az adatbázis kezelésére szolgáló műveletek (lekérdezés, felújítás, új elemek beiktatása stb.).

Legyen adott egy nem elosztott adatbázis a maga adatelemeivel (rekordok, relációk stb.), ezt *logikai adatbázisnak* fogjuk hívni, és egy számítógép-hálózat a maga csomópontjaival és összeköttetéseivel. A logikai adatbázis adatelemeit *logikai elemeknek* (LE) nevezzük. Minden logikai elemből az elosztott adatbázison is csak egyetlen példány létezik. Az adatbázis kezelésére szolgáló műveletek mindig ezen logikai elemekre hivatkoznak. A nem elosztott adatbázisból és a hálózatból úgy hozhatunk létre egy elosztott adatbázist, hogy minden csomóponthoz hozzárendeljük a logikai adatbázis elemeinek egy részhalmazát. Ezenkívül a csomópontokba olyan algoritmusokat telepítünk, amelyek egymással együttműködve biztosítják az adatbázis-kezelő műveletek helyes végrehajtását.

A csomópontokban levő adatelemeket *fizikai adatelemeknek* (*FE*) nevezzük. A logikai adatelemek szétosztásának olyannak kell lennie, hogy

$$\sum LE = \bigcup_{i=1}^{n} \sum FE_i,$$

ahol $\sum LE$ a logikai adatelemek halmaza,

$\sum FE_i$ az i-edik csomópontban levő fizikai adatelemek halmaza.

3.3. DEFINÍCIÓ. *Particionált adatbázis:* $\underset{i \neq j}{\forall} \sum FE_i \cap \sum FE_j = \varnothing$.

3.4. DEFINÍCIÓ. *Teljesen duplikált adatbázis:* $\underset{i,j}{\forall} \sum FE_i = \sum FE_j$.

3.5. DEFINÍCIÓ. *(Részben) duplikált adatbázis:* Minden más esetben.

A logikai adatbázis adatelemeinek értékére bizonyos összefüggéseknek mindenkor érvényben kell lenniük. Ilyen összefüggések pl. egy repülőgép helyfoglaló rendszerben: egy gépre eladott jegyek száma nem haladhatja meg a férőhelyek számát; a lefoglalt helyek számának meg kell egyeznie az eladott jegyek számával. Ezen összefüggéseket *konzisztencia-kritériumoknak* hívják.

3.6. DEFINÍCIÓ. *Konzisztens adatbázis:* Olyan adatbázis, amelyben minden konzisztencia-kritérium igaz.

A nem particionált elosztott adatbázisoknak kétfajta konzisztencia-kritériumnak kell eleget tenniük.

3.7. DEFINÍCIÓ. *Kölcsönös konzisztencia:* Egy logikai adatelem összes fizikai példányának értéke megegyező.

3.8. DEFINÍCIÓ. *Belső konzisztencia:* A logikai adatbázis eleget tesz a konzisztencia-kritériumoknak.

Nincs olyan adatbázis, amely minden időpillanatban konzisztens. A fenti példánál maradva, nem lehet pontosan egyidejűleg végrehajtani a szabad helyek számát nyilvántartó adatelem értékének csökkentését, és az eladott jegyek számát regisztráló adatelem növelését. Az ilyen típusú inkonzisztencia elkerülésére bevezetjük a következő fogalmat:

3.9. DEFINÍCIÓ. *Művelet:* Az adatbázis olyan megváltoztatása, amely véges idő alatt lezajlik és konzisztens állapotú adatbázist konzisztens állapotúba visz át.

Tehát ha az adatbázis a művelet megkezdése előtt konzisztens volt, akkor a művelet befejeződése után is az marad. A továbbiakban az adatbázisnak csak olyan megváltoztatásaival foglalkozunk, amelyek műveletek.

A műveletek tovább bonthatóak *műveleti lépésekre,* de ezek már nem konzisztencia-tartók.

3.10. DEFINÍCIÓ. *Műveleti lépés:* Egy adatelem írása, vagy olvasása.

Egy művelet műveleti lépések sorozatából tevődik össze. Egy, az adatbázis kezelésére szolgáló művelet mindig tartalmaz egy algoritmust, amely az olvasott adatelemek értékeiből és külső információk alapján előállítja az írt adatelemek új értékét. A dolgozatban leírt vizsgálatok szempontjából azonban ezen algoritmusok érdektelenek és ezért a művelet fogalmába sem értjük őket bele. Bennünket egy műveletből csak az olvasott és írt adatelemek halmaza érdekel, mert a különböző műveletek kölcsönhatását ezek alapján vizsgáljuk.

3.11. DEFINÍCIÓ. *Műveletek konfliktusa:* Két műveletet konfliktusban levőnek mondunk, ha az egyik ír olyan adatelemeket, amelyeket a másik is ír, vagy olvas, vagyis az egyik művelet íráshalmazának metszete a másik művelet írás- és olvasáshalmazának uniójával nem-üres.

3.12. DEFINÍCIÓ. *Konkurrens műveletek:* Két művelet akkor konkurrens, ha konfliktusban vannak egymással és az elosztott adatbázison egyidejűleg kerülnek végrehajtásra. Az egyidejűség azt jelenti, hogy az egyik művelet befejezése előtt elindul a másik művelet végrehajtása.

Az adatbázishoz érkező műveletek mindig logikai műveletek, azaz a bennük szereplő írás- és olvasás-halmazok a logikai adatbázisra vonatkoznak. Ha nem tisztán particionált adatbázisról van szó, akkor annak a csomópontnak, ahol a műveletet kezdeményezték, a logikai műveletet fizikai műveletté kell átalakítania. A logikai adatelemekre vonatkozó írásokat meg kell duplikálnia a logikai adatelemekhez tartozó fizikai adatelemeknek megfelelően.

3.13. DEFINÍCIÓ. *Művelet végrehajtása:* A műveleti lépések egymásutáni végrehajtása a fizikai adatelemeken.

Konkurrencia vezérlés

Az elosztott adatbázis csomópontjaiba telepített algoritmusoknak igen sokféle funkciót kell teljesíteniük. Kezelniük kell a helyi adatbázist, a hozzájuk érkező adatbáziskezelő műveleteket részekre kell bontaniuk és a megfelelő csomópontokhoz kell továbbítaniuk stb. Ilyen funkció a konkurrens műveletek vezérlése is.

3.14. DEFINÍCIÓ. *Konkurrens műveletek vezérlése:* Olyan eljárás, amely a konkurrens műveleteknek, vagy műveleti lépéseknek csak olyan végrehajtásait engedi meg, amelyek megtartják az adatbázis konzisztenciáját.

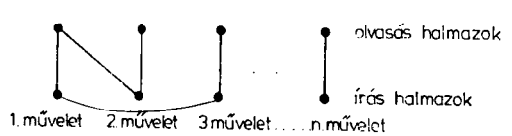
A konkurrencia-vezérlő algoritmusok bizonyos típusai megengedik a műveletek párhuzamos végrehajtását. A műveleti lépések tetszőleges összefésülése esetén nem biztos, hogy az adatbázis konzisztenciája megmarad. Az ilyen típusú algoritmusok konstruálásának egyik fontos eleme a megengedhető összefésülések meghatározása. A csoport tagjait az *adatbázis teljes lezárása nélkül dolgozó algoritmusoknak* hívjuk.

Újabb csoportot alkotnak azok az algoritmusok, amelyek kizárják a párhuzamos művelet-végrehajtást. A csoport tagjait az *adatbázis teljes lezárásával dolgozó algoritmusoknak* hívjuk. A műveletek a folyamatok, a logikai adatelemek az erőforrások. A művelet sikeres végrehajtásához a benne szereplő adatelemek kizárólagos (írás), ill. megosztott (olvasás) tulajdonosának kell lennie.

Konkurrenciavezérlés az adatbázis teljes lezárása nélkül

Az algoritmusok a műveleti lépéseket párhuzamosan hajtják végre; azaz: összefésülik. Ezért az egységnyi idő alatt végrehajtott műveletek száma több, mint a másik csoport algoritmusainál. Ehhez a műveletekről bizonyos előzetes információkat tételeznek fel, történetesen az írás és olvasás halmazaik ismeretét. Mivel ez a legtöbb rendszerben lehetetlen követelmény lenne, a feltétel enyhíthető úgy, hogy e halmazoknak csak egy előre ismert nagyobb halmaz részhalmazának kell lenniük. Természetesen minél kevesebb az előzetes információ, annál kevesebb párhuzamosság engedhető meg. Ad absurdum, ha a műveletekről csak annyi tételezhető fel, hogy az írás és olvasás halmazaik a teljes adatbázis adatelemeinek részhalmazai, akkor ezen algoritmusok sem tesznek mást, minthogy a konfliktusban levő műveleteket sorosan hajtják végre.

Az algoritmusok a konfliktusgráf elemzésen alapulnak. A konfliktusgráfban minden művelethez két csomópont tartozik (1. ábra). Az egyik a művelet olvasás, a másik az írás halmazát reprezentálja. A konfliktusgráfban két csomópontot akkor köt össze él, ha legalább az egyik csomópont írás, és a csomópontok által reprezentált halmazoknak van közös elemük.



1. ábra

A gráf topológiájának vizsgálatával meghatározható, hogy mely műveleteket és milyen mértékben kell szinkronizálni. A műveleti lépések szinkronizálására szolgáló tétel kimondásához néhány újabb definícióval kell megismerkednünk.

3.15. DEFINÍCIÓ. *Jegyzék:* A műveleti lépések egy tetszőleges sorozata.

3.16. DEFINÍCIÓ. *Ekvivalens jegyzékek:* Két jegyzék ekvivalens, ha bennük ugyanazon műveleti lépések szerepelnek és bármely adatbázis állapotra igaz, hogy az egyik jegyzék műveleti lépéseit végrehajtva ugyanazon állapothoz jutunk, mintha a másik jegyzék műveleti lépéseit hajtottuk volna végre.

3.17. DEFINÍCIÓ. *Adatbázis állapot:* Az adatelemek értékeinek összessége.

3.18. DEFINÍCIÓ. *Sorrendi jegyzék:* Nem tartalmaz összefésülést, azaz egy művelethez tartozó műveleti lépések közé egyetlen más művelet műveleti lépése sincs beiktatva.

3.19. DEFINÍCIÓ. *Sorrendezhető jegyzék:* Olyan jegyzék, amelyhez található vele ekvivalens sorrendi jegyzék.

3.1. TÉTEL. Minden olyan szinkronizáció megtartja az adatbázis konzisztenciáját, amely a műveleti lépéseknek csak olyan összefésüléseit engedi meg, amelyek sorrendezhetők.

Bizonyítás. A sorrendi jegyzékek a művelet definíciójából következően megtartják az adatbázis konzisztenciáját. Mivel a sorrendezhető jegyzékeknek mindig van sorrendi ekvivalensük, azért az általuk létrehozott adatbázis-állapotnuk is konzisztensnek kell lennie. A sorrendezhetőség kritériuma a konzisztens szinkronizációnak nem szükséges, csak elégséges feltétele.

A fent vázolt eredményeken alapuló módszerek tipikus képviselője az SDD—1 algoritmus (BERNSTEIN [1, 2, 9]). Mivel az általunk kidolgozott új eljárás nem ezeken alapszik, a továbbiakban nem óhajtunk velük részletesebben foglalkozni.

Konkurrenciavezérlés az adatbázis teljes lezárásával

A csoport minden algoritmusára jellemző, hogy a műveletek végrehajtásában két fázis különböztethető meg. Az egyik az úgynevezett szinkronizációs fázis, amelyben a műveletet végrehajtó folyamat megkísérli mindazon adatelemek birtokba vételét, amelyek a művelet írás- vagy olvasáshalmazához tartoznak. Ha ez sikerült, akkor a művelethez tartozó folyamat a második (végrehajtási) fázisban ténylegesen elvégzi az íráshalmaz adatelemeinek felújítását. A végrehajtási fázist sok esetben megerősítésnek hívják, mert legtöbbször már a szinkronizáció során minden csomópont megkapja az íráshalmaz elemeinek új értékét, így a végrehajtási fázisban a csomópontoknak csak a szinkronizáció sikerességéről kell értesülniük ahhoz, hogy adatbázisukat az új értékekkel felújíthassák.

3.20. DEFINÍCIÓ. *Művelet végrehajtási idő:* Csomóponttól függő. A művelethez tartozó szinkronizáció megkezdésétől az íráshalmaznak a csomóponton való végrehajtásáig tart.

Ha egy végrehajtás alatt álló művelet (M1) valamely csomópontban egy vele konfliktusban álló művelettel (M2) találkozik, akkor a konfliktus feloldására az algoritmusok a következő lehetőségek közül választhatnak:

3.21. DEFINÍCIÓ. *Várakoztatás:* A később érkező (M2) műveletet várakoztatják mindaddig, amíg az M1 művelet végrehajtása be nem fejeződik.

3.22. DEFINÍCIÓ. *Újraindítás:* A később érkező (M2) műveletet törlik (felszámolják) és elölről kezdik a művelet végrehajtását.

3.23. DEFINÍCIÓ. *Legyőzés:* A később érkező (M2) művelet felszámolja az előbb érkezett (M1) műveletet, majd folytatja végrehajtását.

Minden stratégiának megvannak az előnyei és hátrányai. A várakoztatás biztosítja, hogy az egyszer elindított műveletet a rendszer valamikor végrehajtja, de holtpont-veszélyes. Az újraindítás és a legyőzés stratégiájánál nincs holtpont-veszély, de előfordulhat, hogy egy műveletet a rendszer sohasem hajt végre, mert állandóan felszámolja azt (végtelen újraindítás).

Az algoritmusok az adatbázist kétfajta módon, valóságos és virtuális lakatokkal zárhatják le. A valóságos lakatokat a centrális rendszereknél ismertettük. Itt csak annyit teszünk hozzá, hogy ekkor a konfliktusok feloldásának egyedül lehetséges módja a várakoztatás.

3.24. DEFINÍCIÓ. *Virtuális lakat:* Annyiban tér el a valóságos lakattól, hogy feltörhető. Ha egy folyamat lakatját feltörték, akkor a folyamatot fel kell számolni.

A virtuális lakatokkal dolgozó eljárások a konfliktusban álló műveletekre a fent ismertetett stratégiák valamelyikét alkalmazzák attól függően, hogy a konfliktusban szereplő műveleteknek milyen az egymáshoz viszonyított prioritása.

3.25. DEFINÍCIÓ. *Művelet virtuális végrehajtása:* A művelet olvasásainak elvégzése és az írások feljegyzése.

A virtuális műveletet végrehajtó csomópont az írásokat mindaddig megjegyzi, amíg vagy a művelet felszámolásáról, vagy helyes befejezéséről nem értesül. Az előbbi esetben törli a feljegyzett írásokat, az utóbbiban bevezeti a saját adatbázisába (felújítja az adatbázist).

Mielőtt rátérnénk az általunk javasolt algoritmusra, ismertetjük a csoport két képviselőjét. Az algoritmusok helyességének bizonyítása meghaladja ezen dolgozat kereteit, de megtalálható a következő irodalmakban ROSENKRANTZ [8], THOMAS [10].

Rosenkrantz algoritmus

Az algoritmus bármely típusú adatbázison működik.

Az a csomópont, ahol a műveletet kezdeményezték, a művelethez egy egyértelmű azonosítót rendel, majd útjára indítja a hálózatban. A művelet minden olyan csomópontot meglátogat, ahol írni, vagy olvasnivalója van. Az útvonalban egy csomópont többször is szerepelhet. A csomópontbeli adatbázis-kezelők minden látogatáskor virtuálisan végrehajtják a műveletnek azon részét, amely a kezelésük alatt álló adatbázisra vonatkozik. Ha egy műveletnek sikerült a végrehajtásához szükséges összes csomópontot meglátogatnia, akkor az utolsó csomópont a „tedd permanenssé"

üzenetet bocsájtja ki, aminek hatására a csomópontok a műveletet ténylegesen végrehajtják.

A szinkronizációs fázis a kezdeményező csomóponton való virtuális művelet-végrehajtástól az utolsó csomóponton való virtuális művelet-végrehajtásig tart. A megerősítés a „tedd permanenssé" üzenet.

Egy művelet mindig csak egy csomópontban aktív, azaz a műveletet több csomópont párhuzamosan nem hajtja végre.

A művelethez rendelt azonosító egy, a hálózat összes műveletére nézve egyértelmű sorszám, amely a műveleteket lineárisan rendezi.

A módszer virtuális lakatokkal dolgozik. A virtuális lakatok beállítása a műveletnek egy fizikai adatelemhez való hozzáférésekor történik.

A konfliktushelyzetek feloldására a szerzők több lehetőséget is megadnak. Ezek egyike az úgynevezett „várakozás vagy pusztulás" *(wait-die)*. A zárt lakathoz érkező ($M2$) művelet prioritását összehasonlítják a zárat beállító ($M1$) prioritásával:

ha $M2 > M1$: $M2$ várakozik, míg $M1$-et vagy felszámolják, vagy végrehajtják,
ha $M2 < M1$: a rendszer felszámolja $M2$-t.

A módszerről könnyen bizonyítható, hogy holtpontmentes s mivel az adatbázisok teljes lezárásával dolgozik, az engedélyezett jegyzékek sorrendiek.

Thomas (majoritás) algoritmus

Csak teljesen duplikált adatbázison működik, s minden írást az adatelem olvasásának kell megelőznie.

A majoritás-módszer lényege, hogy a virtuális lakatokat elegendő a csomópontok számának felénél eggyel több pontban beállítani (majoritás). Ha minden művelet szinkronizálási fázisához a csomópontok többsége kell, akkor két konkurrens műveletre van legalább egy csomópont, amely még a szinkronizációs fázisban értesül a műveletekről. Ez a csomópont a műveletek prioritását összehasonlítva megakadályozhatja az alacsonyabb prioritású művelet végrehajtását.

A műveletkibocsátó csomópont a műveletet, a kibocsátási időpont és a csomópont sorszámból képzett azonosítóval látja el. A prioritásvizsgálat alapjául ezen azonosító szolgál.

Két konkurrens művelet egy csomópontban a következőképpen találkozhat:

a) mindkét művelet a szinkronizációs fázisban van;
b) az egyik műveletet a vizsgált csomópont már végrehajtotta, a másik a szinkronizációs fázisban van.

A csomópontok az a) esetben a kisebb prioritású művelet, a b) esetben a szinkronizációs fázisban levő művelet végrehajtását kell megakadályoznia. Ezt az algoritmus a következő módon éri el.

A műveletek tartalmazzák az írt/olvasott adatelemeknek a kezdeményező csomópontban levő legutolsó felújítási időpontját (azon művelet kibocsátási időpontját, amely legutoljára az adatelemet írta).

A csomópontokhoz tartozó adatelemek tartalmazzák a legutolsó felújítás időpontját.

A csomópontok minden kapott műveletről szavaznak:

a) a művelet „ELFOGADVA" szavazatot kap, ha a csomópontban nincs várakozó (szinkronizációs fázisban levő) vele konfliktusban álló művelet és az adatbázisban minden, a műveletben szereplő adatelem legutolsó felújítási időpontja a műveleténél öregebb;

b) a művelet „ELUTASÍTVA" szavazatot kap, ha a műveletben szereplő adatelemek között van az adatbázisban a műveleténél fiatalabb;

c) a művelet „TARTÓZKODÁS" szavazatot kap, ha a b) feltétel nem teljesül, de van konfliktusban levő várakozó.

A műveletek terjesztése a hálózatban egy lánc mentén történik. Ha a művelet egyetlen csomópontban is „ELUTASÍTVA" szavazatot kap, akkor a lánc megszakad és a műveletet a rendszer felszámolja. Az „ELFOGADVA" és „TARTÓZKODÁS" szavazatokat adó csomópontok a műveletet virtuálisan végrehajtják. Ha egy művelet a láncban megkapja az „ELFOGADVA" szavazatot a csomópontok több mint felétől, akkor a műveletet a rendszer elfogadta és az utolsó csomópont szétterjeszti a hálózatban a művelet leírását. Ha egy csomópont ilyen üzenetet kap, akkor köteles a benne foglalt műveletet végrehajtani.

Az algoritmus nem tartalmaz várakoztatást, tehát holtpontmentes. A sorrendi jegyzék megtartása a hivatkozott irodalom alapján ellenőrizhető.

4. Egy időzítésen alapuló algoritmus

Javasolt algoritmusunkban az az újszerű, hogy a megerősítést nem üzenetekkel, hanem egy időzítés leteltével oldottuk meg. Ez megbízható hálózat esetén lehetővé teszi, hogy a műveletet minimális üzenetszámmal hajtsuk végre és az időegység alatt végrehajtott műveletek száma maximális legyen.

Első lépésként algoritmusunk működési feltételeit definiáljuk. Ezek szigorú feltételek, de ilyen körülmények között módszerünk nagyon egyszerű. A későbbiekben megmutatjuk, hogy a feltételek az algoritmus bonyolításával hogyan enyhíthetőek, illetve hagyhatóak el.

Feltételek

1. Teljesen duplikált adatbázis;
2. Minden csomóponthoz tartozzék egy óra és ezek legyenek tökéletes szinkronban. Ha $C_i(t)$ az i-edik csomópontban az óra értéke a t időpillanatban, akkor

$$\underset{i,j}{\forall} C_i(t) = C_j(t);$$

3. A műveletek legyenek pillanatszerűek, azaz egy művelet összes olvasása és írása egy időpontban történjék;
4. A hálózat legyen meghibásodásmentes, azaz minden csomópont működjék hibátlanul és minden csomópontból minden csomópont bármely időpillanatban elérhető legyen;
5. A hálózatban legyen meghatározható egy τ időtartam úgy, hogy bármely két csomópont mindig üzenetet válthat τ időn belül.

Egy művelet végrehajtási idejéről

Akármilyen is egy konkurrencia-vezérlő algoritmus, egy művelet végrehajtásához minden csomópontba legalább egy üzenetet el kell juttatni. Az üzenetek eljuttatásához szükséges idő függ a kiadó csomóponttól, az üzenetküldési módtól (üzenetszórás, lánc mentén továbbítás), a hálózat pillanatnyi telítettségétől stb. Nem konstruálható tehát olyan algoritmus, amely tetszőleges hálózaton, minden időpillanatban, minden műveletre minimális végrehajtási időt biztosít. Ami elvárható egy, a művelet végrehajtási ideje szempontjából optimális algoritmustól, az az, hogy ne tartalmazzon olyan megkötöttségeket, amelyek eleve lehetetlenné tesznek egy, az adott környezetben és adott időpillanatban optimális implementációt. Konkrétan, minden csomóponthoz egy és csak egy üzenetet kelljen eljuttatni és az üzenettovábbítási mód tetszőleges lehessen (ne legyen olyan megkötöttség, mint pl. a majoritás-algoritmusnál a láncolt továbbítás).

A műveletek követési távolságáról

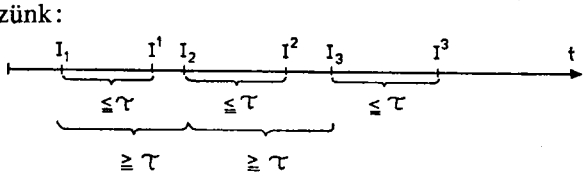
Az adatbázis teljes lezárásával dolgozó algoritmusoknál a konkurrens művelet végrehajtás elkerülésének elégséges feltétele az, hogy a rendszer csak azon műveletet hajtsa végre, amelynek kiadási időpontjában nincs végrehajtás alatt álló, vele konfliktusban levő művelet. Ahhoz, hogy a rendszer eldönthesse, vajon egy művelet kiadási időpontjában van-e konfliktusban levő művelet folyamatban, idő kell, mert vagy meg kell kérdezni a csomópontokat, vagy meg kell várni, amíg a konfliktusban levő műveletek hatása elér a műveletet kezdeményező csomóponthoz. Ha ebben a vonatkozásban is optimális megoldásra törekszünk, akkor a kérdezés rosszabb, mint a várakozás. Várakozni pedig addig kell, amíg a kezdeményező csomóponttól időben legtávolabb eső csomópontból a legrosszabb hálózati körülmények között is megérkezik egy esetleg konfliktusban levő művelet hatása. Ez az időtartam éppen a feltételek között megadott τ.

Az algoritmus elve

A fentiek figyelembevételével egy olyan algoritmust szeretnénk konstruálni, ahol egy művelet végrehajtásához minden csomóponthoz

a) csak egyetlen üzenetet kell eljuttatni;
b) tetszőleges továbbítási móddal;
c) az üzenetek egymást τ időközzel követik.

Ha a műveletek kiadási és végrehajtási időpontját egy időtengely mentén ábrázoljuk, akkor a következő ábrán látható művelet végrehajtási sorozat (jegyzék) elérésére törekszünk:



2. ábra

Az a) feltétel teljesítését csak úgy érhetjük el, ha a megerősítést nem explicit üzenetekkel, hanem pl. egy időzítés lejártával oldjuk meg.

Egy művelet végrehajthatóságának két feltétele van:

$F1$: a műveletet a hálózat minden csomópontja ismerje;
$F2$: a művelet kiadási időpontjában ne legyen folyamatban vele konfliktusban álló művelet.

Ha a kezdeményező csomópont a kiadási időponttal (I_m) egyidejűleg szétterjeszti a műveletet a hálózatban, akkor $F1$ teljesül legkésőbb $I_m+\tau$-ban.

$I_m+\tau$-ban ugyanakkor teljesül $F2$ is, mert az I_m-mel konfliktusban álló műveletek kiadási időpontja $(I_m-\tau, I_m)$ intervallumba esik, tehát $I_m+\tau$-ig ezen üzenetek is megérkeznek minden csomópontba.

A konkurrens műveletek elkerülésére legyen a stratégia a későbbi időpontban kiadott műveletek felszámolása és újraindítása.

A konkurrenciavizsgálatot minden csomópont önállóan végezheti. Egy tetszőleges csomópontban, ha az I_m időpillanatban kiadott üzenetre $I_m+\tau$-ig nem érkezik korábbi időpontú konkurrens művelet, akkor $F1$, $F2$ teljesül és a művelet végrehajtható.

Az előző algoritmusoknál alkalmazott terminológiával egy I_m időpillanatban kiadott üzenetre $I_m+\tau$ időpillanatban a szinkronizáció minden csomópontban befejeződik, és az I_m-ben indított τ időzítés a megerősítés.

Az algoritmus megvalósítása

A műveletet kezdeményező csomópont a művelethez egy időjelzést rendel, amely a helyi óra értéke (I_m) a műveletnek a helyi adatbázison történő „végrehajtásakor". Ezt követően a kezdeményező csomópont a művelettel ugyanúgy jár el, mint ahogyan azt egy tetszőleges csomópontra a következőekben leírjuk. Ha a vizsgálatok során a műveletet nem vetette el, akkor szétterjeszti azt a hálózatban (időjelzés — mely egyben azonosító —, az olvasott és írt adatelemek értéke).

A) Ha egy csomópont kap egy műveletleírást, akkor a következőképpen viselkedik:

1. Ha nincs a csomópontban várakozó művelet, amely konfliktusban van a kapott művelettel, akkor a csomópont a kapott műveletet várakozónak tekinti és elindít egy $\tau-(I_h-I_m)$ időzítést — ezt időzítés 1-nek hívjuk —, ahol I_h a helyi óra értéke az üzenet megkapásának időpillanatában.

2. Ha a csomópontnak van várakozó művelete, amely konfliktusban van a kapott művelettel, akkor ha

(i) $I_m<I_v$ (I_v a várakozó művelet kiadási időpontja), akkor az I_v-hez tartozó műveletet eldobja és I_m-mel 1. szerint jár el;

(ii) $I_v+\tau>I_m>I_v$, akkor az I_m műveletet eldobja;

(iii) $I_m>I_v+\tau$, akkor I_m műveletével 1. szerint jár el.

B) Ha egy csomópontban egy időzítés1 lejár:
a műveletet végrehajtja és elindít egy újabb τ időtartamú időzítést — ezt időzítés2-nek hívjuk.

C) Ha egy csomópontban egy időzítés2 lejár:
a műveletet nem tekinti várakozónak.

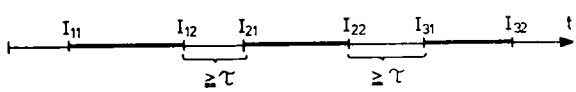
A második időzítés indítása az első pillanatban feleslegesnek tűnhet. Bár nem felesleges, de nem is az egyedüli megoldása a következő implementációs problémának: minden I_m-re minden csomópontnak ellenőriznie kell, volt-e $I_m-\tau$-ban kiadott, az I_m-ben kiadott művelettel konfliktusban levő művelet. Lehetnek olyan csomópontok, amelyek az I_m-ben kiadott művelet megkapásának időpontjában egy vele konfliktusban álló műveletet már végrehajtottak. Ezért egy műveletet a végrehajtás után még legalább τ ideig meg kell őrizni a későbbiekkel való összehasonlítás végett.

A pillanatszerű művelet feltétel elhagyása

Ha a műveleteket valamilyen nem nulla, de véges idejűnek tekintjük, akkor az előző ábra a következőképpen módosul: (a műveletek végrehajtási időpontját az egyszerűség kedvéért nem tüntettük fel)

Az I_{i1} az i-edik művelet első adathozzáférési időpontja, I_{i2} a művelet „befejeződésének" időpontja (az az időpont, amikor a művelet terjesztésre kerülhet).

4.1. TÉTEL. Ha I_{m2}-t tekintjük egy művelet kiadási időpontjának, akkor I_m helyett I_{m2}-t használva a vizsgálatok változatlanok maradhatnak, ha a műveletet kiadó csomópont az $F3$ feltétel teljesülését biztosítja.



3. ábra

$F3$: egy művelet csak akkor kerül terjesztésre, ha végrehajtásának I_{m1} és I_{m2} időpontja között a csomópont egyetlen vele konfliktusban álló műveletről sem értesül.

Bizonyítás. A hálózatban bármely két konfliktusban álló műveletre igaz: $I_{i2}-I_{j1}<\tau$ (negatív is lehet) $\rightarrow I_{i2}-I_{j2}<\tau$, mert $F3$ biztosítja, hogy ha egy művelet konfliktusban van egy korábbival és mégis terjesztésre kerül, az csak úgy lehetséges, hogy a terjesztés időpontjában a csomópont még nem értesült az előző műveletről. Az előzőekből következik:

$$I_{i2}-I_{j2}>\tau \rightarrow I_{i2}-I_{j1}>\tau \quad \text{implikáció.}$$

A feltétel igaz voltát az algoritmus $I_m=I_{m2}$ helyettesítéssel biztosítja.

A meghibásodásmentes hálózat feltétel elhagyása

A hálózatról a meghibásodásmentesség helyett a következőket tételezzük fel: bármely csomópont egy üzenet elküldésétől számított ϱ időn belül észleli, hogy az üzenetet a címzett nem kapta meg.

Egészítsük ki az algoritmusunkat azzal, hogy bármely csomópont, ha úgy találja,

hogy az általa elküldött üzenetet a címzett nem kapta meg, akkor a hálózat minden csomópontja felé egy, a műveletet felszámoló üzenetet küld. Ilyen üzenet vétele esetén a csomópont a hivatkozott műveletet nem hajtja végre.

Ahhoz, hogy egy esetleges felszámoló üzenetet minden csomópont még a művelet végrehajtása előtt megkapjon, algoritmusunkban τ helyett mindenütt $\tau+\varrho+\tau$-t kell használnunk. Ekkor a következő végrehajtási képet kapjuk:

4. ábra

A felszámoló üzenet bevezetésével is akadhatnak csomópontok, akik a műveletet végrehajtják (akik a művelet vétele után, de még a felszámoló üzenet előtt leszakadnak a hálózatról). Ha a hálózat nem minden csomópontja egységesen hajt végre, vagy vet el egy műveletet, akkor az adatbázisban inkonzisztencia keletkezik. Mivel a hálózat meghibásodása esetén az inkonzisztenciát elkerülni nem tudjuk, ezért biztosítani kell, hogy a hálózat helyreállítása után az adatbázis az utolsó, még helyesen végrehajtott műveletnek megfelelően helyreállítható legyen.

Ehhez egészítsük ki algoritmusunkat a következőkkel:

a) egy csomópont felfüggeszti a műveletek végrehajtását, ha egy felszámoló üzenetet kapott vagy generált, mindaddig, amíg a hálózat helyreállításáról nem értesül;
b) minden csomópont az általa végrehajtott műveletek azonosítójáról egy feljegyzést vezet.

Az a) kiegészítés biztosítja, hogy egy hibás hálózaton előbb-utóbb minden csomópont felfüggeszti a műveletek végrehajtását. Egy csomópont ugyanis akkor nem észleli a hálózat meghibásodását, ha a felszámoló üzenet nem érkezik meg hozzá. Ehhez egy újabb meghibásodásnak kell keletkeznie. Tehát egy hibás hálózaton egy csomópont maximálisan addig hajthat végre műveleteket, amíg egy meghibásodás olyan közel nem kerül hozzá, hogy önmaga észleli. A csomópontok számának és a hálózat topológiájának ismeretében minden csomópontra meghatározható az a maximális műveletszám, amit a csomópont a hiba keletkezése és észlelése között végrehajthat. Ezen maximumok maximuma legyen K. K-ra egy, a topológiától független felső becslés $(n(n-1))/2$, ahol n a csomópontok száma.

A b) kiegészítés az összeköttetések megjavítása után az adatbázis konzisztenciájának helyreállításához szükséges. Egy hibátlan hálózatban a feljegyzések tökéletesen azonosak. Akkor kezdenek különbözni, amikor a meghibásodás miatt egyes csomópontok „önálló" életet kezdenek élni. A hálózatban mindig van legalább egy csomópont, amely a hálózat első meghibásodását észlelte, tehát az ő feljegyzésében szereplő műveletek még mind a hibátlan hálózaton hajtódtak végre. A hálózat helyreállítása után meg kell keresni azt a feljegyzést, amely minden más feljegyzésnek része, s a hozzá tartozó csomópont adatbázisa alapján a teljes adatbázis helyreállítható.

A feljegyzések hossza, azaz hogy a csomópontoknak az utolsó hány műveletet kell megjegyezniük, az előző bekezdésben foglaltak alapján meghatározott K.

Ha algoritmusunkban a meghibásodást figyelembe akarjuk venni, akkor a $\tau=2\tau+\varrho$ helyettesítés mindenképpen szükséges. A továbbiak csupán egy lehetséges megoldást vázolnak fel annak szemléltetésére, hogy az algoritmus nem megbízható hálózaton is alkalmazható. Természetesen más megoldások is léteznek, s ezek a megfontolások más helyzetben is alkalmazhatóak.

A tökéletes szinkron feltétel elhagyása

A $C_i(t)=C_j(t)$ feltétel helyett vezessük be a valóságos órákra elérhető $\forall_{i,j} |C_i(t)-C_j(t)|<\varepsilon$-t.

A probléma abból adódik, hogy a csomópontok a saját és a kibocsátó csomópont órájának pontatlansága miatt csak bizonytalanul tudják meghatározni azt az időintervallumot, ami a művelet tényleges kibocsátása óta eltelt. Az algoritmusban az időzítésnek kettős szerepe van: egyrészt biztosítja, hogy egy műveletről a végrehajtásakor már minden csomópont értesült, másrészt nem enged meg konkurrens műveletvégrehajtást. Ahhoz, hogy algoritmusunkat módosíthassuk, pontosan megfogalmazzuk az órákkal kapcsolatos feltételezéseinket.

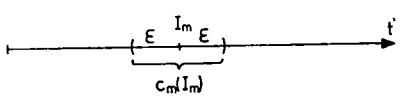
$F4$: $\forall_{i,j} |C_i(t)-C_j(t)|<\varepsilon.$

$F5$: $\sum_{i=1}^{n} C_i(t)/n=t+\Delta,$

ahol Δ egy nem feltétlenül pozitív, additív tag a valóságos fizikai időhöz képest, és azon intervallumban $(2-3\tau)$, ahol vizsgálatainkat végezzük, nem változik.

Bár algoritmusunkban mindenütt időpontok különbsége szerepel és ezért Δ konkrét értéke érdektelen, a pontosság kedvéért az elkövetkezőkben mindenütt $t'=t+\Delta$-val számolunk.

4.2. TÉTEL. Egy műveletnek a $c_m(I_m)$ időjelzése a tényleges kibocsátási időpontnak (I_m) ε sugarú környezetébe esik.



5. ábra

Bizonyítás. Az $F5$ feltétel biztosítja, hogy a tényleges kibocsátási időpont mindig a legkisebb és legnagyobb óraérték közé esik. Az $F4$ feltétel szerint ezen szélső értékek maximális eltérése kisebb, mint ε, tehát egy tetszőleges csomópontban a valóságos idő (t'-ben mért) és a helyi óra értéke nem térhet el jobban egymástól, mint ε.

4.3. TÉTEL. Egy műveletnek két különböző csomópontban való végrehajtása nem különbözik egymástól jobban, mint ε.

Bizonyítás. A $C_m(I_m)$ időjelzésű művelet érkezzék az i-edik csomópontba I_i, a j-edik csomópontba I_j időpillanatban.

Az i-edik csomópontban a művelet időzítése lejár: $\tau-[c_i(I_i)-c_m(I_m)]+I_i$ időpillanatban.

A j-edik csomópontban a művelet időzítése lejár: $\tau-[c_j(I_j)-c_m(I_m)]+I_j$ időpillanatban.

A kettő különbségének abszolút értéke:

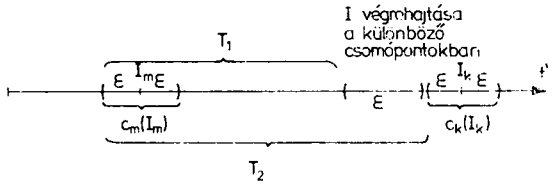
$$|\{\tau-[C_i(I_i)-C_m(I_m)]+I_i\}-\{\tau-[C_j(I_j)-C_m(I_m)]+I_j\}| =$$

$$= |C_i(I_i)-C_j(I_j)+I_j-I_i| = |C_i(I_i)-C_j(I_i+\Delta t)+I_i+\Delta_t-I_i| =$$

$$= |C_i(I_i)-C_j(I_i)-C_j(\Delta t)+\Delta t| < \varepsilon.$$

Mivel $|C_i(I_i)-C_j(I_i)|<\varepsilon$ és a Δt idő alatt az órák tökéletesen pontosnak tekinthetőek, ezért $C_j(\Delta t)=\Delta t$.

Ezen tételek ismeretében a következő ábra a műveletek megengedhető végrehajtását szemlélteti t' függvényében:



6. ábra

T_1 értékét úgy kell megválasztani, hogy annak leteltére a $C_m(I_m)$ időjelzésű műveletet minden csomópont megkapja és értesüljön az összes vele konfliktusban álló korábbi időjelzésű műveletről. Ha T_1 értékét $\tau+\varepsilon$-ra választjuk, akkor az I_m tényleges kibocsátási időpont és az első végrehajtás között biztos, hogy τ, vagy annál hosszabb idő telik el. A T_1-et követő ε intervallumban az adatbázis nem feltétlenül konzisztens, tehát az előzővel konfliktusban álló műveletet csak ezen intervallum után engedhetjük meg. Ehhez T_2 értékét $\tau+2\varepsilon$-ra kell választani.

Valóságos órák esetén algoritmusunkat tehát a következőképpen kell módosítani:

a) az időzítést τ helyett $\tau+\varepsilon$-ra kell változtatni;
b) az összehasonlításokban τ helyett $\tau+2\varepsilon$-t kell használni.

A teljesen duplikált adatbázis feltétel enyhítése

A feltétel a következőképpen enyhíthető: minden művelethez a kezdeményező csomópontban álljon rendelkezésre a teljes olvasás-halmaz.

Ha ezt biztosítjuk, akkor a műveletek a kezdeményező csomópontban tulajdonképpen „végrehajthatóak". Azaz, a műveletek olvasás halmaza alapján azok írás

halmazát elő tudják állítani. A szétterjesztés csak a helyi adatbázisokba való bevezetés miatt szükséges. Eddigi algoritmusunkkal szemben annyi változás történik, hogy a csomópontok a művelet íráshalmazának csak azon elemeit hajtják végre, amelyeket adatbázisuk tartalmaz. Az üzenetek száma sajnos nem csökkenthető, mert minden csomópontnak minden műveletről tudnia kell (olyanokról is, amelyeknek egyetlen elemét sem hajtja végre), hogy a csomópontok azonos műveleteket fogadjanak, illetve vessenek el.

5. Következtetések

Dolgozatunkban ismertettük a folyamatok szinkronizálásának problémáját, alapfogalmait és néhány algoritmusát centrális és elosztott rendszerekben. Megadtuk az elosztott adatbázisok konkurrens folyamat vezérlésének (szinkronizációjának) egy új algoritmusát, amely az időzítésen alapszik. Az algoritmusnak a többiekkel szembeni előnye a művelet végrehajtásához szükséges minimális üzenetszám és végrehajtási idő, a csekély segédinformáció. Hátránya, hogy nem minden adatbázison alkalmazható, s van egy viszonylag nehezen implementálható eleme: az időzítés.

Az időzítéssel kapcsolatban meg kell jegyeznünk, hogy az algoritmusban szereplő τ időintervallumot nem szükséges a hálózat által megengedett minimumra választani. Bár így az időegység alatt végrehajtható műveletek száma csökken, de nagyobb τ esetén egyrészt az időzítés kvantálható (implementációs egyszerűsítés), másrészt a hálózat rövid idejű meghibásodása nem eredményezi τ túllépését és ezzel a rendszer leállását, majd újraindulását.

Az algoritmus egyéb feltételei a modern rendszerekben általában könnyen teljesíthetőek.

A számítógépekben alkalmazott órák pontossága 10^{-7}, 10^{-8} nagyságrendű, tehát a feltételek minden külön szinkronizáció nélkül τ-nál akár egy nagyságrenddel kisebb ε-ra is hosszú időn keresztül fennállnak. Ezen felül az órák üzenetekkel szinkronizálhatóak. Ennek pontosságára L. LAMPORT [6] végzett vizsgálatokat és állapított meg összefüggéseket. Az ő eredményei itt is alkalmazhatóak, pusztán az szükséges, hogy az óraszinkronizáló üzenetek továbbítása a hálózatban lényegesen pontosabban történjék, mint az egyéb üzeneteké.

IRODALOM

[1] BERNSTEIN, P. A., ROTHNIE, J. B., GOODMAN, N. and PAPADIMITRIOU, C. A., "The concurrency control mechanism of SDD-1: A system for distributed databases (the fully redundant case)", *IEEE Transaction on Software Engineering* **SE-4** (1978) 154–167.

[2] BERNSTEIN, P. A., SHIPMAN, D. W. and ROTHNIE, J. B., "Concurrency control in a system for distributed databases (SDD-1)", *ACM Transaction on Database Systems* (1980) 18–51.

[3] DIJKSTRA, E. W., "Solution of a problem in concurrent programming control", *Communications of the ACM* **8** (1965) 569.

[4] HABERMANN, A. W., "Prevention of system deadlocks", *Communications of the ACM* **12** (1969) 373–377.

[5] LAMPORT, L., "A new solution of Dijkstra's concurrent programming problem", *Communications of the ACM* **17** (1974) 453–455.

[6] LAMPORT, L., "Time, clocks and the ordering of events in a distributed system", *Communications of the ACM* **21** (1978) 558–565.

[7] RICART, G. and AGRAWALA, A. K., "An optimal algorithm for mutual exclusion in computer networks", *Communications of the ACM* **24** (1981) 9–17.
[8] ROSENKRANTZ, D. I., STEARNS, R. E. and LEWIS, P. M., "System level concurrency control for distributed database systems", *ACM Transactions on Database Systems* (1978) 178–198.
[9] ROTHNIE, J. B., BERNSTEIN, P. A. *et al.*, "Introduction to a system for distributed databases (SDD–1)", *ACM Transactions on Database Systems* (1980) 1–17.
[10] THOMAS, R. H., "A majority consensus approach to concurrency control for multiple copy databases", *ACM Transactions on Database Systems* (1979) 180–209.

(Beérkezett: 1981. október 8.)

POGÁNY ANDRÁS
MTA SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET
1132 BUDAPEST, VICTOR HUGO U. 18.

CONCURRENCY CONTROL IN DISTRIBUTED DATABASES, A TIMING BASED ALGORITHM

A. POGÁNY

This paper is dealing with the problem of process synchronization in centralised and distributed systems. Some fundamental definitions are given first and a few algorithms for synchronization in centralised systems are discussed. When dealing with distributed systems databases and their concurrency control mechanisms are concerned. The strictly serial algorithms of THOMAS and ROSENKRANTZ et. al. are analysed next. Finally a new algorithm is proposed for distributed concurrency control which uses timing instead of messages for confirmation of transactions. By using timing mechanism the synchronization of transactions with minimal number of messages is resolved. In this sense the suggested algorithm is optimal. Some implementation problems of the algorithm as e.g. real clocks, recovery etc. are detailed at the end of the paper.

Alkalmazott Matematikai Lapok 7 (1981) 331—344

ABSZTRAKT OSZTOTT ADATTÍPUSOK EGY SPECIFIKÁCIÓJA

KOZMA LÁSZLÓ

Budapest

A cikkben módszert közlünk absztrakt osztott adattípusok egy lehetséges specifikálására. A módszer elemzése során bevezetjük a konzisztens specifikáció fogalmát. Megvizsgáljuk a konzisztens specifikáció szemantikáját és módszert adunk a specifikáció holtpont-mentességének eldöntésére.

A specifikációs módszert egy példán keresztül mutatjuk be.

1. Bevezetés

Az adatabsztrakció jelentőségét soros programokra már régen felismerték a szakemberek. Az absztrakt adattípusok hasznos eszközei a megbízható szoftvert előállító módszereknek. Napjainkban már szinte elképzelhetetlen, hogy az absztrakció eszközei nélkül sikeresen létrehozhatók legyenek nagy összetett programrendszerek. Az ilyen programrendszerek ugyanis korunkban annyira bonyolultakká váltak — nem utolsó sorban a megoldandó feladat összetettsége, nagysága miatt —, hogy áttekintésük, megértésük is igen komoly feladatot jelent, nem beszélve karbantartásukról. Az absztrakció lényege éppen abban áll, hogy eszközt ad a programok bonyolultságának csökkentésére, a programok különálló részekre való bontásával. Emellett lehetőséget ad egy adott objektum lényeges jellemzőinek szétválasztására azoktól, amelyek az adott környezetben lényegtelenek.

Az absztrakció még inkább hasznos eszköze lehet párhuzamos programok specifikálásának. Az adatok megosztása az egymással konkurráló folyamatok között ugyanis mind a programok áttekinthetőségét tovább rontja, mind a programok helyességének bizonyítását nagyon komplikálttá teszi.

Egy *absztrakt adattípus* absztrakt objektumok egy osztályát specifikálja az objektumokon végrehajtható műveletek halmazával együtt.

Egy *absztrakt osztott adattípus* absztrakt objektumok olyan osztályát specifikálja, amelyet az egymással konkurráló folyamatok közösen használhatnak az objektumok műveleteinek párhuzamos végrehajtása útján.

Az absztrakt osztott adattípus specifikációja így az objektumok és a rajtuk értelmezett műveletek specifikációja mellett a szinkronizáció specifikációját is kell, hogy tartalmazza. A szinkronizációra mind az objektumok belső konzisztenciájának megőrzése, mind a magasabb szintű szinkronizációs döntések szempontjából szükség van.

Az utóbbi időben egyre több szakember foglalkozik az adatabsztrakcióra vonatkozó kutatásokkal. Ennek egyrészt az az oka, hogy a szoftver előállításának költsége viszonylag megnőtt a hardver költségekhez képest, a hardver technika rohamos fej-

lődése miatt. Másrészt ugrásszerűen megnőtt az igény a nagy programrendszerek készítése iránt, aminek eredményeként a szoftver előállításának költségei is megnövekedtek. Gondolunk itt elsősorban a programok tesztelésére, hibák megkeresésére, kijavítására, mint legjelentősebb költségtényezőkre. Mindezek hatására gazdasági szempontból megnőtt a megbízható szoftver előállítását célzó kutatások jelentősége. Ezzel egyidőben intellektuális szempontból is felértékelődtek — a feladat nehézsége, összetettsége miatt — az ilyen irányú kutatások.

Az eddigiek során jelentős erőfeszítések történtek az absztrakt adattípusok specifikálása terén GUTTAG [6], LISKOV és ZILLES [14], LISKOV és BERZINS [13], WULF és társai [20]. Ugyanakkor számos nyelvet fejlesztettek ki az adatabsztrakció támogatására, például a CLU LISKOV és társai [15] vagy az ALPHARD SHAW és társai [17] nyelveket. Kidolgoztak az adatabsztrakcióra vonatkozó bizonyítási módszereket is HOARE [8], SPITZEN és WEGBREIT [18], VARGA [19], WULF és társai [21] stb. Az absztrakt adattípusok hasznos építőkövei a jól strukturált párhuzamos programoknak is OWICKI [16]. LAVENTHAL kidolgozott egy módszert absztrakt adattípusok szinkronizációs tulajdonságainak specifikálására LAVENTHAL [12]. Ez a módszer lehetőséget ad arra, hogy a szinkronizációt függetlenül specifikáljuk az absztrakt adattípus többi ún. „adatkapcsolatú" tulajdonságától. A szinkronizáció—specifikáció ilyen megközelítési módja számos előnnyel jár, melyek közül talán a legfontosabb a modularitás biztosítása, ami nemcsak a specifikáció, hanem a majdani implementálás során is megtartható. Ilyen módon, ha egy soros (nem konkurrens) specifikáció létezik egy absztrakt adattípusra vonatkozóan, akkor a specifikációhoz egyszerűen hozzávéve a szinkronizáció specifikációját nyerhetjük az osztott adattípus teljes specifikációját.

Tanulmányunkban megadunk egy módszert absztrakt osztott adattípusok specifikálására, amely a következő elemeket tartalmazza: az „adatkapcsolatú" részek specifikálására *Hoare axiomatikus rendszerét* használjuk HOARE [8], a szinkronizáció specifikálására pedig a LAVENTHAL által kidolgozott specifikációs nyelvet LAVENTHAL [12].

A további fejezetekben röviden ismertetjük a szinkronizáció—specifikáció módszerét (második fejezet), megadjuk az absztrakt osztott adattípusok specifikációját (harmadik fejezet), elemezzük specifikációs módszerünket (negyedik fejezet), majd egy példát közlünk absztrakt osztott adattípusok specifikálására (ötödik fejezet).

2. A szinkronizáció specifikációja

A konkurrens folyamatok közös céljuk megvalósítása érdekében általában azonos erőforrásokat használnak fel. A folyamatoknak tehát együtt kell működniük erőforrásaik használata során. Ezt az együttműködést valamilyen módon koordinálni kell. A szinkronizáció célja éppen — legáltalánosabb értelemben — a koordináció megvalósítása. A koordináció egyik fajtája például az, amikor korlátozzuk az egyidőben egy adott erőforrást használó folyamatok számát. Egy másik fajta koordináció, amikor bizonyos ütemezési feladatokat kell megoldani, például a sorbanállási problémák esetében.

A párhuzamos programozási nyelvek kapcsán eddig számos nyelvi primitívet dolgoztak ki a szinkronizációs feladatok megoldására. Ilyen nyelvi eszköz például

a szemafor DIJKSTRA [5]. A problémakör kitűnő összefoglalását BRINCH HANSEN [2] tanulmányában találhatjuk meg. Ezek a nyelvi eszközök alkalmasak a szinkronizációs feladatok megoldására, de nem alkalmasak a specifikáció céljaira, mivel még az olyan jól strukturált eszközök, mint a feltételes kritikus szakasz utasítások BRINCH HANSEN [1], vagy a monitorok HOARE [9] alkalmazása esetén is a szinkronizáció kifejezése igen bonyolult. Még ezen eszközök esetében is igen nagy a fogalmi szakadék a szinkronizációról alkotott intuitív fogalmunk és a megvalósítás eszközei között.

LAVENTHAL kidolgozott egy eszköztárat a szinkronizáció specifikálására LAVENTHAL [12]. Ezen eszközök felhasználásával a szinkronizációs problémákat természetesebb módon fejezhetjük ki. A specifikációs módszer alapját egy nyelv képezi, amelyen a szinkronizációs követelmények megfogalmazhatók.

A modell, amelyen a specifikációs nyelv alapul, feltételezi, hogy a specifikációt megvalósító kód csak bizonyos események osztályaival kapcsolatos. Egy adatobjektum minden műveletéhez időben három pontot — eseményt — kapcsolunk: egy ún. *request*, egy *enter* és egy *exit* eseményt. Egy művelethez tartozó *request* esemény akkor következik be. amikor az objektumot használó folyamat közli a szinkronizációs mechanizmussal a művelet végrehajtására vonatkozó igényét. Ezt követi az az időpont, amikor a szinkronizációs mechanizmus megengedi a művelet megkezdését, azaz bekövetkezik a művelethez tartozó *enter* esemény. Végül a művelet végrehajtása után a folyamat közli a szinkronizációs mechanizmussal a művelet befejeződésének tényét — bekövetkezik az *exit* esemény. Ezek az események az adattípus minden p műveletére vonatkozóan a p^{request}, p^{enter}, ill. p^{exit} eseményosztályokat alkotják. Ezekre az eseményosztályokra vonatkozó idő szerinti korlátozásokat kell a szinkronizációs kódnak megvalósítania, s ezt a szinkronizációs kódot kell tudnunk specifikálni a specifikációs nyelv eszközei segítségével.

2.1. *A szinkronizáció-specifikáció nyelve*

A szinkronizáció-specifikáció nyelve az elsőrendű predikátumkalkuluson alapul. Ezt a kalkulust egészítjük ki azzal a lehetőséggel, hogy egyrészt hivatkozhassunk az objektum egyes műveleteinek aktivizálásához rendelt eseményekre, másrészt, hogy definiálhassuk az idő szerinti rendezést az események között.

Jelölje $p[j]$ egy objektum p műveletének j-edik aktivizálását, ahol j az ún. aktivizálási sorszám az objektum teljes történetére vonatkozóan. A $p[j]$ aktivizáláshoz tartozó *request, enter* és *exit* eseményeket jelölje rendre $p[j].request$, $p[j].enter$, ill. $p[j].exit$. Az egy adott objektumosztályhoz tartozó összes ilyen esemény időben teljesen rendezett. Jelölje ezt a rendezési relációt „$\rightarrow$", melynek jelentése: „időben megelőzi".

Például a

$$p[j].enter \rightarrow q[i].exit$$

reláció, rendezési klóz, azt a tényt fejezi ki, hogy a p művelet j-edik aktivizálásának *enter* eseménye időben megelőzi a q művelet i-edik aktivizálásának *exit* eseményét.

Egy absztrakt adattípusra vonatkozó szinkronizáció-specifikáció a fenti rendezési klózokból és esetleg a rájuk vonatkozó argumentum korlátozásokból a normál predikátum-kalkulus operátoraival előállított formula.

A továbbiakban feltesszük, hogy az aktivizálási számok jelölésére használt i, j változók a fenti klózokban minden lehetséges értékre vonatkozóan univerzálisan kvantáltak.

A nyelv pontos szintaxisát és szemantikáját LAVENTHAL [12] adja meg.

3. Absztrakt osztott adattípusok egy specifikációja

Egy absztrakt osztott adattípus absztrakt objektumok olyan osztályát specifikálja, amelyet az egymással konkurráló folyamatok egyidőben közösen használhatnak az objektumok műveleteinek párhuzamos végrehajtásával. Az adattípusok specifikálásának egy lehetséges módja az, hogy szétválasztjuk az adatkapcsolatú részek specifikációját a szinkronizáció specifikációjától. Ebben az esetben az adatkapcsolatú részeket bármely soros esetre kidolgozott módszerrel specifikálhatjuk, így például a CLU nyelv „cluster"-eivel LISKOV és társai [15], vagy a *Hoare-féle axiomatikus módszer* segítségével HOARE [8]. A továbbiakban egy olyan specifikációs módszert adunk meg, ahol az adatkapcsolatú részeket a *Hoare-féle axiomatikus módszerrel* specifikáljuk, a szinkronizáció specifikálására pedig a Laventhal által kidolgozott nyelv jól-formált formuláit használjuk fel.

Egy absztrakt osztott adattípus egy lehetséges specifikációja a következő lehet:

type típusnév ($\bar{p}$):

elvárás: Elvárás ($\bar{p}$)

kezdeti feltétel: Kezd(A)

invariáns: $I_a(A)$

műveletek:

műveletnév (result $\bar{x}$; $\bar{y}$)

$pre_a(A', \bar{x}', \bar{y}')$

$post_a(A, \bar{x}, \bar{y})$

⋮

szinkronizáció: $S(A)$

ahol

$\bar{p}$: az objektum paramétereinek listája,
A: az absztrakt objektum,
$\bar{x}$: a művelet kimenő paraméterei,
$\bar{y}$: a művelet bemenő paraméterei,

$A', \bar{x}', \bar{y}'$ jelöli rendre az absztrakt objektumot és a paraméterek értékeit a műveletek megkezdése előtt.

A specifikáció során első lépésként megadjuk az új absztrakt típus nevét és a típusban specifikált absztrakt objektumot.

Az elvárás klóz — Elvárás — egy állítás, amely az absztrakt adattípus paramétereire ír elő korlátozásokat. A típus egy újonnan létrehozott példánya csak akkor használható korrektül, ha az új példány paraméterei kielégítik az Elvárás állítást.

A kezdeti feltétel — Kezd — egy állítás, amely a típus egy új példányára megadja a kezdeti objektumot, amelyből az absztrakt műveletek alkalmazásával az absztrakt objektumok minden további példánya generálható.

Az invariáns klóz — $I_a(A)$ — egy konzisztens állítás, amely az absztrakt objektumok adott osztályát definiálja.

Egy művelet specifikációja megadja a művelet nevét, formális paramétereit és azok típusát. Az előfeltétel — pre_a — egy állítás, amely megadja azt a feltételt, amelynek teljesülése esetén a művelet végrehajtása korrekt. Az utófeltétel — $post_a$ — egy állítás, amely leírja a művelet hatását, feltéve, hogy a művelet megkezdése előtt a pre_a előfeltétel teljesült. Ez azt jelenti, hogy a $post_a$ állítás a következő relációt jelöli: $(A, \bar{x}) = f_a(A', \bar{x}', \bar{y}')$, ahol f_a írja le az absztrakt művelet hatását. Vagyis az absztrakt művelet az absztrakt objektum egy új példányát hozza létre és értéket ad az eredmény paramétereknek.

A szinkronizáció — $S(A)$ — egy jól formált formula a szinkronizáció-specifikáció nyelvén, azaz az $S(A)$ formula az egyes absztrakt műveletekhez tartozó eseményosztályokra vonatkozó idő szerinti korlátozásokat specifikálja a teljes absztrakt adattípusra vonatkozóan.

4. A specifikációs módszer elemzése

4.1. *A konzisztens specifikáció fogalma*

Az absztrakt osztott adattípus előző fejezetben megadott specifikációs módja lehetővé teszi, hogy szétválasszuk az adatokkal kapcsolatos részek specifikálását a szinkronizációs tulajdonságok specifikálásától. Ez azonban nem jelenti azt, hogy a külön-külön specifikált két rész teljesen független egymástól. Ellenkezőleg, az absztrakt objektumok belső konzisztenciájának megőrzése miatt a specifikációs eszközöknek olyan erősnek kell lenniük, hogy valahányszor egy f_a absztrakt művelet végrehajtása előtt teljesül az $S(A)$ szinkronizációs előírás, mindannyiszor teljesüljön az f_a művelet pre_a előfeltétele is.

DEFINÍCIÓ: Egy absztrakt osztott adattípus specifikációja *konzisztens,* ha minden f_a absztrakt műveletére teljesül a következő: valahányszor az f_a absztrakt művelet végrehajtása előtt teljesül az $S(A)$ szinkronizációs előírás, mindannyiszor teljesül az f_a művelet pre_a előfeltétele is.

A konzisztencia a specifikáció igen fontos tulajdonsága. Ha egy absztrakt osztott adattípus specifikációja nem konzisztens, akkor vagy a szinkronizációs előírásokat kell erősíteni — további rendezési klózokat kell megadni — vagy a műveletek specifikációját kell megváltoztatni úgy, hogy a konzisztenciát biztosító gyengébb előfeltételek is elegendőek legyenek az utófeltételek teljesüléséhez.

Tekintsünk egy egyszerű példát, amely bemutatja konkurrens környezetben a konzisztencia fontosságát: tegyük fel, hogy adott egy korlátos absztrakt puffer

osztott specifikációja egy *append,* egy *remove* és egy *empty* művelettel. Az *append* művelettel egy újabb elemet tehetünk a pufferbe, ha nincs tele. A *remove* művelettel egy elemet vehetünk ki a pufferből, ha az nem üres. Az *empty* pedig egy olyan logikai függvény (művelet), amely megadja, hogy a puffer üres-e vagy sem. Tegyük fel, hogy a specifikáció nem konzisztens, például a *remove* műveletre vonatkozóan, azaz valahányszor a *remove* végrehajtása előtt teljesül a szinkronizáció-specifikáció, ebből nem következik, hogy mindannyiszor a *remove* előfeltétele — a puffer nem üres — is teljesül. Ilyen esetben egy folyamat az *empty* logikai függvénnyel lekérdezheti ugyan, hogy a puffer nem üres-e, de pozitív válasz esetén sem biztos, hogy mire ugyanaz a folyamat végrehajtja a *remove* utasítást, a pufferben még mindig van elem; ugyanis közben más folyamatok kivehették az összes elemet a pufferből. Konzisztens specifikáció esetén ilyen nem fordulhat elő.

A továbbiakban csak konzisztens specifikációkkal foglalkozunk.

4.2. *Absztrakt osztott adattípusok szemantikája*

Egy absztrakt osztott adattípus tulajdonságai az ún. aktor modell alapján tárgyalhatók. A modell elméleti alapjait IRENE GREIF dolgozta ki GREIF [4]. Az aktor modell segítségével az adattípusok szemantikájának ún. esemény-orientált megközelítését adhatjuk meg, mivel az aktor modell lényege, hogy a kiszámítási rendszereket események halmazával jellemzi.

Egy egyszerű szekvenciális folyamat például az aktor modellben események teljesen rendezett halmazával jellemezhető, ahol az események közötti rendezést — amit a $\Rightarrow$ jelöl — az események egymásutáni szekvenciális végrehajtásával definiáljuk. A $\Rightarrow$ rendezést a továbbiakban szekvenciális rendezésnek nevezzük.

Például legyen P egy olyan szekvenciális folyamat, amely az $\{E_1, \ldots, E_n\}$ események halmazából áll és az események közötti szekvenciális rendezés $E_1 \Rightarrow E_2 \Rightarrow \ldots \Rightarrow E_n$.

A P eszerint egy olyan szekvenciális folyamat, amely először az E_1 eseményt hajtja végre, majd a rákövetkező E_2-t és így tovább, legvégül az E_n eseményt hajtja végre.

Párhuzamos folyamatokból álló rendszerek az aktor modellben az események parciálisan rendezett halmazával jellemezhetők. A folyamatok párhuzamos végrehajtásából eredő különböző magasabb szintű szinkronizációs problémák leírására az ún. idő szerinti rendezés vezethető be. E rendezés jele a $\rightarrow$, aminek jelentése: időben megelőzi.

DEFINÍCIÓ: Legyenek adottak az $S_1 = \{E_1^1, \ldots, E_n^1\}$ és $S_2 = \{E_1^2, \ldots, E_m^2\}$ folyamatok az $E_1^1 \Rightarrow \ldots \Rightarrow E_n^1$, ill. $E_1^2 \Rightarrow \ldots \Rightarrow E_m^2$ szekvenciális rendezésekkel. Az S_1 folyamat időben megelőzi S_2-t, azaz

$$S_1 \rightarrow S_2, \quad \text{ha} \quad E_1^1 \Rightarrow \ldots \Rightarrow E_n^1 \rightarrow E_1^2 \Rightarrow \ldots \Rightarrow E_m^2.$$

Legyen adott egy x absztrakt osztott adattípus, $q_1, \ldots, q_k$ műveletekkel. Az adattípus egy q_j $(1 \leqq j \leqq k)$ műveletének jelentését események egy R_{q_j} sorozata definiálja a következő módon:

$$R_{q_j} = \{E_{q_{j_{req}}}, E_{q_{j_{ent}}}, E_{q_{j_1}}, \ldots, E_{q_{j_m}}, E_{q_{j_{ex}}}\},$$

ahol

$$E_{q_{j_{req}}} \Rightarrow E_{q_{j_{ent}}} \Rightarrow E_{q_{j_1}} \Rightarrow \ldots \Rightarrow E_{q_{j_m}} \Rightarrow E_{q_{j_{ex}}}$$

és a $\Rightarrow$ rendezés a szekvenciális rendezés,

$$E_{q_{j_{req}}} = q_j[i].request$$

$$E_{q_{j_{ent}}} = q_j[i].enter$$

$$E_{q_{j_{ex}}} = q_j[i].exit \text{ alakú események.}$$

Az $E_{q_{j_1}}, \ldots, E_{q_{j_m}}$ események a q művelet törzsének végrehajtása során aktivált, az absztrakt objektumra vonatkozó események. Konzisztens specifikáció esetén a q_j műveletet az $E_{q_1}, \ldots, E_{q_{j_m}}$ események pontos megadása helyett jellemezhetjük a pre_{q_j}, ill. $post_{q_j}$ elő- és utófeltételek megadásával.

Az egyes q_j műveletek eseményei közötti idő szerinti rendezést az x absztrakt osztott adattípusra az $S(x)$ szinkronizáció specifikáció adja meg.

Mivel az $S(x)$ csak az $E_{q_{j_{req}}}, E_{q_{j_{ent}}}, E_{q_{j_{ex}}}$ típusú események között specifikálja a rendezést, így elegendő az egyes q_j műveleteket olyan absztrakciós szinten tekinteni, ahol

$$R'_{q_j} = \{E_{q_{j_{req}}}, E_{q_{j_{ent}}}, E_{q_{j_{ex}}}\} \quad \text{és} \quad E_{q_{j_{req}}} \Rightarrow E_{q_{j_{ent}}} \Rightarrow E_{q_{j_{ex}}}$$

továbbá konzisztens specifikáció esetén tudjuk, hogy az $E_{q_{j_{ex}}}$ esemény bekövetkezésekor teljesül a $post_{q_j}$ állítás. Így konzisztens specifikáció esetén az x absztrakt adattípus szemantikáját azzal a kiszámítási történettel definiáljuk, ami érvényes $S(x)$-re és amelyben az események $\forall_j$-re $1 \leqq j \leqq k$ az R'_{q_j} halmazból valók, továbbá $\forall_j$-re $1 \leqq j \geqq k$ az $E_{q_{j_{ex}}}$ bekövetkezésekor teljesül a $post_{q_j}$ állítás x-re, mivel az $E_{q_{j_{ent}}}$ esemény bekövetkezése előtt teljesült a pre_{q_j} előfeltétel.

Így az x absztrakt osztott adattípus szemantikáját konzisztens specifikáció esetén visszavezethetjük a szinkronizáció-specifikáció nyelvének szemantikájára. A nyelv pontos, esemény-orientált szemantikáját LAVENTHAL [12] tartalmazza.

4.3. *Holtpontmentes specifikáció*

Párhuzamos folyamatokból álló programok tulajdonságainak vizsgálata esetén egyik alapvető kérdés, hogy a programok holtpontmentesek-e vagy sem COFFMANN és társai [3], HABERMANN [7], HOLT [10] stb. Ugyanez a kérdés felvethető az absztrakt adattípusok osztott specifikációjára vonatkoztatva is.

R. C. HOLT a holtpontot úgy definiálta, mint olyan helyzetet, ahol egy vagy több folyamat végtelen ideig blokkolt állapotba kerül, mivel erőforrás-igényeik sohasem teljesíthetők. Absztrakt adattípusokra vonatkoztatva e definíció azt jelenti, hogy holtpont akkor következhet be, ha a szinkronizáció-specifikáció „túl sok megszorítást" tartalmaz az események sorrendjére vonatkozóan, s emiatt az események valamely csoportja sohasem következhet be. Matematikai szempontból azonban a „túl sok megszorítás" fogalma nem elég pontos. Egyáltalán egy adott specifikációról hogyan tudjuk eldönteni, hogy „túl sok megszorítás"-t tartalmaz-e vagy sem? A továbbiakban e kérdéskör vizsgálatával fogunk foglalkozni.

Legyen adott egy x absztrakt osztott adattípus specifikációja. Az adattípus műveleteit jelölje $q_1, \ldots, q_m$. Az egyes műveletek aktiválásai során végrehajtott események $\forall_i$-re, $\forall_k$-ra $(1 \leqq k \leqq m)$

$$q_k[i] \,.\, request \Rightarrow q_k[i] \,.\, enter \Rightarrow q_{k_1} \Rightarrow \ldots \Rightarrow q_{k_n} \Rightarrow q_k[i] \,.\, exit$$

alakúak.

Tudjuk, hogy az $S(x)$ szinkronizáció-specifikáció csak a *request, enter* és *exit* típusú események között ír elő idő szerinti korlátozásokat. Feltehetjük továbbá, hogy ha egy $q_k[i].enter$ esemény bekövetkezett, akkor a $q_{k_1}, \ldots, q_{k_n}$ események minden további korlátozás nélkül véges időn belül bekövetkeznek. Így a holtpont vizsgálata szempontjából elegendő a műveleteket olyan absztrakciós szinten tekinteni, ahol egy q_k művelet i minden lehetséges értékére

$$R_{q_k}[i] = \{q_k[i] \,.\, request,\ q_k[i] \,.\, enter,\ q_k[i] \,.\, exit\}$$

alakú események halmaza, amelyre a

$$q_k[i] \,.\, request \Rightarrow q_k[i] \,.\, enter \Rightarrow q_k[i] \,.\, exit$$

szekvenciális rendezés érvényes. Az $S(x)$ szinkronizáció-specifikáció az ilyen $R_{q_k}[i]$ alakú halmazokra ír elő idő szerinti korlátozásokat. Modellünkből továbbá következik, hogy $\forall k$-ra a q_k művelet minden i és j aktiválására $q_k[i].request \rightarrow q_k[j].request$ ún. természetes korlátozás teljesül, ha $i<j$.

DEFINÍCIÓ: Egy folyamat egy x absztrakt adattípus q_k műveletének végrehajtásakor blokkolt állapotba kerül egy e esemény aktiválása előtt, ha az $S(x)$ specifikációban előírt, az e eseményt megelőző valamely esemény még nem következett be.

DEFINÍCIÓ: Két vagy több folyamat egy x absztrakt adattípus műveleteinek végrehajtása közben holtpontba kerül, ha mindegyik folyamat blokkolt állapotban van és a várakozó (blokkolt) folyamatok ciklikus láncot alkotnak.

DEFINÍCIÓ: Egy x absztrakt adattípus specifikációja holtpontmentes, ha az x adattípuson osztozó konkurrens folyamatok sohasem kerülhetnek holtpontba.

Kérdés: eldönthető-e pusztán az $S(x)$ szinkronizáció-specifikáció ismeretében, hogy az x absztrakt adattípuson osztozó konkurrens folyamatok holtpontba kerülhetnek-e vagy sem? Hogy erre a kérdésre válaszolhassunk, bevezetjük a szinkronizáció-specifikáció gráf modelljét.

A továbbiakban feltesszük, hogy az $S(x)$ szinkronizáció-specifikáció nem tartalmaz argumentumokra vonatkozó korlátozásokat. Vegyük az $S(x)$ szinkronizáció-specifikációt és hozzuk diszjunktív normál formára.

Jelölje a szinkronizáció-specifikáció így nyert alakját $S'(x) = D_1 \vee D_2 \vee \ldots \vee D_n$, ahol $\forall k$-ra $(1 \leqq k \leqq n)$ D_k rendezési klózok vagy azok negáltjának konjunkciója. $\forall k$-ra $(1 \leqq k \leqq n)$ D_k-nak feleltessük meg a következő G_k gráfot: ha valamely $q_{m_1}[i].et_1 \rightarrow q_{m_2}[j].et_2$ rendezési klóz D_k-nak egy konjunkciós tagja, ahol $et_1, et_2 \in \{request, enter, exit\}$, akkor a $q_{m_1}[i].request$, $q_{m_1}[i].enter$, $q_{m_1}[i].exit$, ill. $q_{m_2}[j].request$, $q_{m_2}[j].enter$, $q_{m_2}[j].exit$ események legyenek a G_k gráf új csomópontjai, ha a G_k gráfban ezek a csomópontok még nem szerepeltek. A G_k gráf új élei legyenek a $(q_{m_1}[i].request, q_{m_1}[i].enter)$, $(q_{m_1}[i].enter, q_{m_1}[i].exit)$, $(q_{m_2}[j].request, q_{m_2}[j].enter)$, $(q_{m_2}[j].enter, q_{m_2}[j].exit)$ irányított élek, ha még a gráfban nem sze-

repeltek. (Megjegyzés: (a, b) jelöli az a csomópontból a b csomópontba mutató irányított élet.)

A fenti élek az adott események közötti szekvenciális rendezéseket reprezentálják.

Új élként vegyük hozzá a G_k gráfhoz a $(q_{m_1}[i].et_1, q_{m_2}[j].et_2)$ irányított élet, amely az adott rendezési klózt reprezentálja.

A G_k gráf a D_k-ban szereplő összes rendezési klózra tartalmazza a fenti módon a csomópontokat és az éleket. Megjegyezzük, hogy ha D_k egy rendezési klóz negáltját tartalmazza, $\neg(q_{m_1}[i].et_1 \rightarrow q_{m_2}[j].et_2)$, akkor a G_k gráfhoz a szekvenciális rendezésből eredő élek mellett új élként a $(q_{m_2}[j].et_2, q_{m_1}[i].et_1)$ irányított élet vesszük hozzá. Végül vegyük hozzá G_k-hoz az ún. természetes korlátozásból származó éleket.

Lényeges kiemelni, hogy a fenti eljárással annyi különböző G_k gráfot kapunk, ahány D_k diszjunkciós tag szerepel az $S'(x)$ specifikációban.

TÉTEL: Adott x absztrakt adattípuson osztozó konkurrens folyamatok $\Leftrightarrow$ kerülhetnek holtpontba, ha az adattípus $S'(x)$ szinkronizáció-specifikációjához tartozó G_k gráfok közül legalább az egyik tartalmaz irányított kört.

KÖVETKEZMÉNY: Egy adott x absztrakt adattípus specifikációja $\Leftrightarrow$ holtpontmentes, ha az $S'(x)$ szinkronizáció-specifikációnak megfelelő G_k gráfok egyike sem tartalmaz irányított kört.

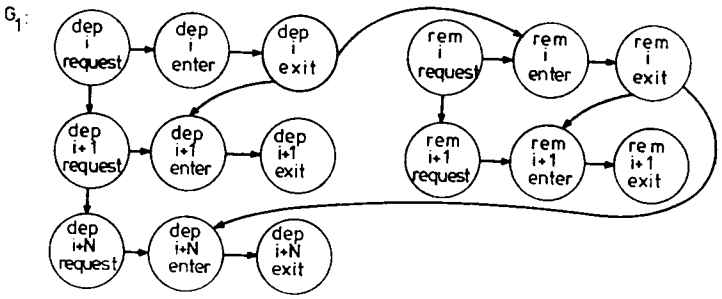
Bizonyítás: Az adott x absztrakt adattípus q_j műveleteihez tartozó $q_j[i].request$, $q_j[i].enter$, $q_j[i].exit$ események $\forall i$-re, $\forall j$-re időben teljesen rendezettek. Így egy adott esemény bekövetkezése előtt mindig eldönthető, hogy az adott eseménnyel kibővített kiszámítási résztörténet érvényes-e az $S'(x)$ szinkronizáció-specifikációra vonatkozóan, vagy sem. Rendszerünk olyan, ami csak az érvényes kiszámítási résztörténeteket engedi meg. Mivel az egyes eseményeket az absztrakt adattípus műveleteit végrehajtó folyamatok aktiválják a végrehajtás különböző fázisaiban, így a request, enter és exit események közötti szekvenciális rendezéseket is kielégíti minden kiszámítási résztörténet. Ilyen módon az egyes D_k diszjunkciós tagokhoz rendelt G_k gráfok irányított élei által reprezentált precedencia relációk a kiszámítási történet egyes lehetséges megengedett résztörténeteit specifikálják. Ezért nyilvánvaló, hogy két vagy több folyamat $\Leftrightarrow$ kerülhet holtpontba, ha valamelyik G_k gráf irányított hurkot tartalmaz.

Példa. Tekintsük a korlátos absztrakt osztott puffer szinkronizáció-specifikációját:

$$
\begin{aligned}
D_1\colon\ & dep\,[i]\,.\,exit \rightarrow rem\,[i]\,.\,enter \wedge \\
& rem\,[i]\,.\,exit \rightarrow dep\,[i+N]\,.\,enter \wedge \\
& dep\,[i]\,.\,exit \rightarrow dep\,[i+1]\,\,.\,enter \wedge \\
& rem\,[i]\,.\,exit \rightarrow rem\,[i+1]\,\,.\,enter
\end{aligned}
$$

ahol a *dep* művelettel egy elemet tehetünk be a pufferbe, ha nincs tele; a *rem* művelettel egy elemet vehetünk ki a pufferből, ha nem üres.

A D_1 specifikációnak megfelelő G_1 gráfot az 1. ábra tartalmazza. A G_1 gráfról leolvasható, hogy a specifikáció holtpontmentes, ha $N>0$. $N=0$ esetén a $(rem\,[i].exit,\ dep\,[i+0].enter)$ irányított él létrejöttével kialakul a $dep\,[i].enter$, $dep\,[i].exit$, $rem\,[i].enter$, $rem\,[i].exit$ csomópontok alkotta irányított kör.



1. ábra

5. Korlátos absztrakt osztott verem specifikációja

Célunk egy olyan verem specifikálása, amelyen egyidőben több folyamat osztozhat. Mindegyik folyamat tetszőlegesen tehet be elemeket a verembe, ill. vehet ki onnan elemeket. A verem mérete véges.

type stack (n: integer):

stack = sequence (a)

elvárás $n > 0$

kezdeti feltétel nullsequence (a_0)

invariáns $0 \leqq$ length (stack) $\leqq n$

műveletek

push (s: stack, y: elem)

pre: $0 \leqq \text{length}(s) < n$ post: $s = s' \sim y$

pop (s: stack, result y)

pre: $0 < \text{length}(s) \leqq n$ post: $y = \text{last}(s') \wedge s = \text{leader}(s')$

szinkronizáció:

$$
\begin{aligned}
S(\text{stack}):\ & push[i].exit \rightarrow pop[i].enter \wedge \\
& pop[i].exit \rightarrow push[i+n].enter \wedge \\
& push[i].exit \rightarrow push[i+1].enter \wedge \\
& pop[i].exit \rightarrow pop[i+1].enter \wedge \\
& (push[i].exit \rightarrow pop[k].enter \vee \\
& pop[k].exit \rightarrow push[i].enter)
\end{aligned}
$$

Megjegyzések:

a) sequence $(a)=\langle a_1, \ldots, a_k\rangle$ a specifikált elemek sorozata, speciálisan a $\langle\,\rangle$ jelöli az üres sorozatot, amelyet „nullsorozat"-nak nevezünk.

b) $s\sim y$ jelöli azt a sorozatot, amelyet az s sorozat és az y elem konkatenációjával nyerünk.

c) length (s) jelöli az s sorozat hosszát.

d) last (s) jelöli az s sorozat utolsó, legjobboldalibb elemét.

e) leader (s) jelöli azt a sorozatot, amit úgy kapunk az s sorozatból, hogy töröljük az s utolsó elemét.

f) A szinkronizáció-specifikáció első konjunkciós tagja biztosítja, hogy a *pop* művelet üres vermen nem dolgozhat. A második tag előírja, hogy a *push* művelet tele verembe nem tehet sohasem újabb elemet. A következő két tag azt fejezi ki, hogy két *push*, ill. két *pop* művelet egyszerre nem kerülhet végrehajtásra. Az utolsó tag biztosítja, hogy egy *push* és egy *pop* művelet is kölcsönösen kizárják egymást.

5.1. *A specifikáció elemzése*

Állítás: A korlátos absztrakt osztott verem specifikációja konzisztens.

Bizonyítás: Azt kell bizonyítanunk, hogy valahányszor egy *push*, ill. *pop* művelet végrehajtása előtt teljesül a szinkronizáció előírása, mindannyiszor teljesül a *push*, ill. *pop* művelet előfeltétele is.

A bizonyítást a *push*, ill. *pop* műveletek aktivizálási számaira vonatkozó ún. párhuzamos indukcióval végezhetjük el. Először belátjuk, hogy $i=0$-ra teljesül állításunk, majd feltesszük, hogy minden $0\leqq k\leqq j<i$-re teljesül állításunk és belátjuk, hogy akkor i-re is teljesül, ahol j a push műveletek aktiválási száma, k pedig a *pop* műveleteké.

A) Tegyük fel, hogy $i=0$, azaz még nem történt meg egyetlen *push*, ill. egyetlen *pop* művelet aktiválása sem. Ekkor teljesül a nullsequence (s_0) kezdeti feltétel, ebből és a length függvény definíciójából következik, hogy length $(s_0)=0$. Mivel az elvárás klóz alapján $n>0$, így $0\leqq$length $(s_0)<n$ teljesül az első *push* művelet aktivizálása előtt. A *pop* művelet $0<$length $(s_0)\leqq n$ előfeltétele nem teljesül, de ekkor a szinkronizáció-specifikáció első tagja sem teljesül, mivel *push* [1].*exit* → *pop* [1]. .*enter* rendezési klóz csak akkor teljesülhet, ha az első *push* művelet végrehajtása már befejeződött. A *push* művelet végrehajtása után viszont az utófeltételéből és a *length* függvény definíciójából következik, hogy

$$n\geqq\text{length}\,(s) = \text{length}\,(s_0\sim x) = \text{length}\,(s_0)+1=0+1>0,$$

vagyis teljesül a *pop* művelet előfeltétele is.

B) Tegyük fel, hogy a tétel állítása minden olyan *pop*, ill. *push* műveletre igaz, amelyeknek aktiválási száma k, ill. j, ahol $0\leqq k\leqq j<i$. Belátjuk, hogy akkor az i-edik *push*, ill. *pop* műveletre is igaz az állítás.

Feltevésünk azt jelenti, hogy $\forall j$-re $0\leqq j<i$, a j-edik *push* művelet végrehajtása után $0\leqq$length $(s)=$length $(s')+1\leqq n$ teljesül. Illetve $\forall k$-ra $0<k\leqq j<i$ a k-adik *pop* művelet végrehajtása után $0\leqq$length $(s)=$length $(s')-1\leqq n$ teljesül. Mivel a specifikáció szerint az egyes műveletek kölcsönösen kizárják egymást, így $\forall k$-ra, $\forall j$-re, $0\leqq k\leqq j<i$, length $(s)=j-k$.

A továbbiakban két esetet kell megkülönböztetnünk aszerint, hogy az i-edik *push* vagy az i-edik *pop* műveletet vizsgáljuk-e.

B/1. Tegyük fel, hogy $j=i-1$, azaz az $(i-1)$-edik *push* művelet befejeződött. Az indukciófeltevés szerint az $(i-1)$-edik *push* művelet előfeltétele teljesült. Belátjuk, hogy valahányszor a szinkronizáció-specifikáció teljesül az i-edik *push* művelet előtt, mindannyiszor teljesül a *push* művelet előfeltétele is.

Az $(i-1)$-edik *push* művelet végrehajtása után length $(s)=i-1-k$, ahol $i-1-n \leqq k \leqq i-1$ szükségképpen teljesül az indukciófeltevés alapján k-ra. Tegyük fel, hogy $k=i-1-n$, azaz az $(i-1-n)$-edik *pop* művelet fejeződött még csak be.

Ekkor length $(s)=i-1-(i-1-n)=n$, vagyis a *push* művelet előfeltétele nem teljesül, de akkor nem teljesül a szinkronizáció-specifikáció második tagja sem, mely előírja, hogy az i-edik *push* művelet aktiválása csak akkor következhet be, ha az $(i-n)$-edik *pop* művelet végrehajtása befejeződött, azaz

$$pop\,[i-n]\,.\,exit \rightarrow push\,[i]\,.\,enter.$$

Az $(i-n)$-edik *pop* művelet a specifikáció szerint bekövetkezhet és rá érvényes az indukciófeltevés. Belátható továbbá, hogy $\forall k$-ra $i-n \leqq k \leqq i-1$ esetén a *push* művelet előfeltétele mindig teljesül, így teljesül akkor is, ha a szinkronizáció-specifikáció teljesül. Ugyanis

$$0 \leqq \text{length}\,(s)=i-1-(i-n) = n-1<n, \quad \text{ha} \quad k=i-n,$$

$$\text{ill.} \quad 0 \leqq \text{length}\,(s) = i-1-(i-1) = 0<n, \quad \text{ha} \quad k=i-1.$$

B/2. Tegyük fel, hogy a $k=(i-1)$-edik *pop* művelet befejeződött, ekkor az eddigiekhez hasonlóan, beláthatjuk, hogy ha az i-edik *pop* művelet végrehajtása előtt teljesül a szinkronizáció-specifikáció, akkor teljesül az i-edik *pop* művelet előfeltétele is.

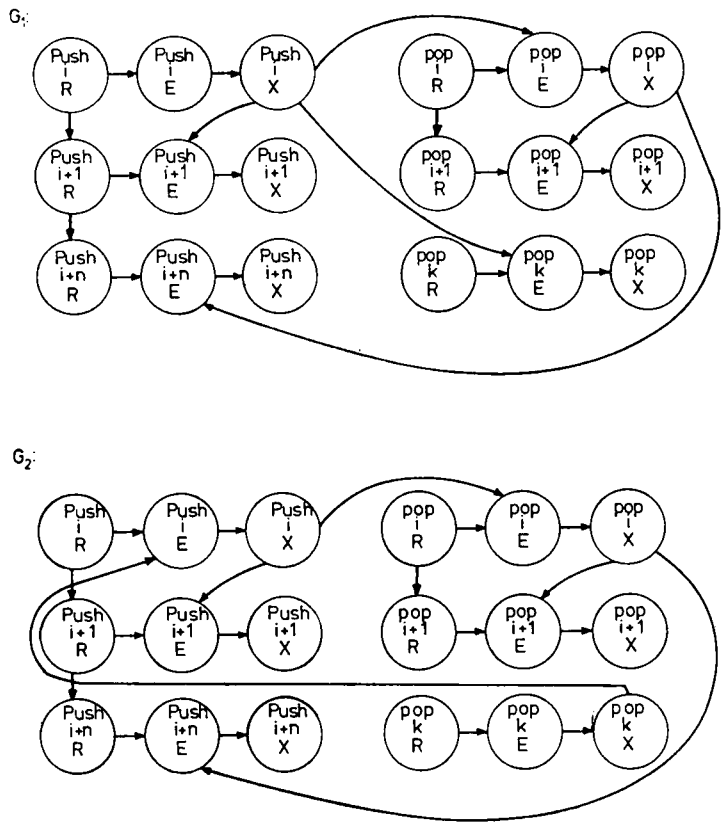
5.2. *Holtpontmentesség*

Hozzuk az S (stack) szinkronizáció-specifikációt diszjunktív normál formára:

S' (stack):

$$\begin{aligned}
&(push\,[i]\,.\,exit \rightarrow pop\,[i]\,.\,enter \wedge \\
&\quad pop\,[i]\,.\,exit \rightarrow push\,[i+n]\,.\,enter \wedge \\
&\quad push\,[i]\,.\,exit \rightarrow push\,[i+1]\,.\,enter \wedge \\
&\quad pop\,[i]\,.\,exit \rightarrow pop\,[i+1]\,.\,enter \wedge \\
&\quad push\,[i]\,.\,exit \rightarrow pop\,[k]\,.\,enter) \\
\vee\ &(push\,[i]\,.\,exit \rightarrow pop\,[i]\,.\,enter \wedge \\
&\quad pop\,[i]\,.\,exit \rightarrow push\,[i+n]\,.\,enter \wedge \\
&\quad push\,[i]\,.\,exit \rightarrow push\,[i+1]\,.\,enter \wedge \\
&\quad pop\,[i]\,.\,exit \rightarrow pop\,[i+1]\,.\,enter \wedge \\
&\quad pop\,[k]\,.\,exit \rightarrow push\,[i]\,.\,enter)
\end{aligned}$$

A D_1 és D_2 diszjunkciós tagoknak megfelelő G_1 és G_2 gráfot a 2. ábra tartalmazza. A gráfokról leolvashatjuk, hogy $n>0$ esetén a specifikáció holtpontmentes.



2. ábra

6. Konklúzió

A tanulmányban közölt specifikációs módszer alkalmas absztrakt adattípusok specifikálására párhuzamos programozási környezetben. Megvizsgáltuk a specifikáció legfontosabb tulajdonságait, így a specifikáció holtpontmentességét, valamint bevezettük a konzisztens specifikáció fogalmát. A specifikáció implementálása és az implementálás helyességének kérdéskörével itt nem foglalkoztunk, mivel az meghaladta jelen tanulmányunk kereteit.

IRODALOM

[1] BRINCH HANSEN, P., "A comparison of two synchronizing concepts", *Acta Informatica* **1** (1972) 190–199.
[2] BRINCH HANSEN, P., "Concurrent programming concepts", *Computing Surveys* **5** (1973).
[3] COFFMANN, E. G., ELPHICK, M. J. and SHOSHANI, A., "Systems deadlocks", *Computing Surveys* **3** (1971).
[4] GREIF, I., *Semantics of Communicating Parallel Processes* (M.I.T., 1975).
[5] DIJKSTRA, E. W., *Cooperating Sequential Processes, Programming Languages* (Academic Press, New York, 1968).
[6] GUTTAG, J. V., The specification and application to programming of abstract data types" Computer Science Ph. D. Thesis, University of Toronto, 1975.
[7] HABERMANN, A. N., "Prevention of systems deadlocks", *Communications of the ACM* **12** (1969) 373–377, 385.
[8] HOARE, C. A. R., "Proof of correctness of data representations", *Acta Informatica* **1** (1972) 271–281.
[9] HOARE, C. A. R., "Monitors: An operating system structuring concept", *Communications of the ACM*, **17** (1974) 549–556.
[10] HOLT, R. C., On deadlocks in computer systems, Ph. D. Thesis, Cornell University, Ithaca, N. Y., 1971.
[11] KOZMA, L., „Absztrakt adattípusok specifikációja párhuzamos programozási környezetben", Programozási Rendszerek '81 Konferencia, Szeged.
[12] LAVENTHAL, M. S., Synthesis of synchronization code for data abstractions, M. I. T. Laboratory for Computer Science, 1978.
[13] LISKOV, B. H. and BERZINS, V., *An Appraisal of Program Specifications* (Computation Structures Group, Memo 141, M. I. T., 1976).
[14] LISKOV, B. H. and ZILLES, S., "Specification techniques for data abstractions", *IEEE Transactions on Software Engineering* **SE–1** (1975) 7–19.
[15] LISKOV, B. H., SNYDER, A., ATKINSON, R. and SCHAFFART, C., "Abstraction mechanism in CLU", Communications of the ACM, **20** (1977) 564–576.
[16] OWICKI, S., An axiomatic proof technique for parallel programs, II. Shared data abstractions, Stanford University, 1976.
[17] SHAW, M., WULF, W. A. and LONDON, R. L., "Abstraction and verification in Alphard", *Communications of the ACM* **20** (1977) 553–564.
[18] SPITZEN, J. and WEGBREIT, B., "The verification and synthesis of data structures", *Acta Informatica* **4** (1975) 127–144.
[19] VARGA, L., "Specification of reliable software", *MTA SZTAKI Tanulmányok* **113** (1980) 309—325.
[20] WULF, W., LONDON, R. L. and SHAW, M., "An introduction to the construction and verification of Alphard programs", *IEEE Transactions on Software Engineering* **SE–2** (1976) 253–264.
[21] WULF, W. A., LONDON, R. L. and SHAW, M., "Abstraction and verification in Alphard: A symbol table example", in: Proceedings of IFIP TC2 Working Conference, Novoszibirszk, 1977.

(Beérkezett: 1981. szeptember 21.)

KOZMA LÁSZLÓ
SZÁMÍTÓGÉPALKALMAZÁSI KUTATÓ INTÉZET
1015 BUDAPEST, CSALOGÁNY U. 30—32.

A SPECIFICATION OF SHARED ABSTRACT DATA TYPES

L. KOZMA

A method is presented for specifying shared abstract data types. The notation of consistent specification is introduced and the properties of the specification method are discussed.
A simple technique is given for detecting deadlock contained in the specification. Some examples demonstrate the proposed techniques.

Alkalmazott Matematikai Lapok 7 (1981) 345—356

A KÉMIAI REAKCIÓK MATEMATIKAI MODELLJEINEK KONZISZTENCIÁJÁRÓL[1]

L. ARNOLD

Bréma

1. Bevezetés

Térbeli tartományban végbemenő kémiai reakciókat két fő elv alapján szoktak modellezni:

(i) *globális* (azaz diffúzió nélküli, térben homogén, vagy „tökéletesen kevert") vagy *lokális* (azaz diffúziót figyelembe vevő, térben inhomogén) leírásukat adják,

(ii) *determinisztikus* (makroszkopikus, fenomenologikus, koncentrációkat felhasználó) vagy *sztochasztikus* (a részecskeszámok szintjén történő, a belső fluktuációkat figyelembe vevő) leírásukat végzik el. Ennek a két elvnek a kombinálása lényegében négy különböző matematikai modellt ad, lásd például: HAKEN [7] és NICOLIS és PRIGOGINE [15].

Ez a dolgozat röviden leírja ezeket a modelleket, továbbá azzal a kérdéssel foglalkozik, hogy ezek a modellek konzisztensek-e egymással, és ha igen, akkor milyen (matematikai) értelemben.

Mivel inkább a fogalmakra koncentrálunk, mintsem a teljes általánosságot célozzuk meg, ezért arra az esetre szorítkozunk, amikor csak egyetlen olyan X reaktáns van jelen, amelynek a koncentrációja nincs állandó szinten tartva, s amikor egy egydimenziós $\Omega \subset \mathbf{R}^1$ térrészben zajlik az alábbi típusú sémával megadott reakció:

$$\text{(1)} \qquad \sum_{j=1}^{r} \alpha_{ij} A_j + \beta_i X \underset{k_i'}{\overset{k_i}{\rightleftharpoons}} \sum_{j=1}^{r} \alpha_{ij}' A_j + (\beta_i + 1) X,$$

$$\alpha_{ij}, \alpha_{ij}', \beta_i \in \mathbf{N} = \{0, 1, 2, \ldots\}; \quad k_i, k_i' \geqq 0, \quad i = 1, \ldots, s.$$

2. Kémiai reakciók matematikai modelljei

2.1. *A globális determinisztikus modell*

Az (1) reakciósémáról leolvasható a

$$\lambda(x) = \sum_{i=1}^{s} \alpha_i k_i x^{\beta_i} = \sum_{i=0}^{p} a_i x^i, \quad a_i \geqq 0,$$

[1] Ez a dolgozat fordítása a következőnek: ARNOLD, L., "On the consistency of the mathematical models of chemical reactions", in: *Dynamics of Synergetic Systems* Ed. H. HAKEN (Springer Verlag, Berlin–Heidelberg–New York, 1980), 107–118. A fordítás közléséhez a szerző hozzájárult, a Copyright tulajdonosa, a Springer Verlag pedig térítés nélkül engedélyezte a fordítást.

keletkezési tag, és a

$$\mu(x) = \sum_{i=1}^{s} \alpha_i' k_i' x^{\beta_i+1} = \sum_{i=1}^{q} b_i x^i, \quad b_i \geqq 0$$

megsemmisülési tag.

Így az X reaktáns $\varphi=\varphi(t)$, $t\in[0, \infty)$ koncentrációjának fejlődését a

$$\frac{d\varphi(t)}{dt} = f(\varphi(t)), \quad \varphi(0) = \varphi_0 = 0, \tag{2}$$

$$f(x) = \lambda(x)-\mu(x) = \sum_{i=0}^{d} c_i x^i, \quad c_0 \geqq 0, \quad c_d \neq 0$$

nemlineáris közönséges differenciálegyenlet (kinetikai egyenlet) írja le.

Ha $c_d<0$, akkor (2) megoldása minden t-re definiálva van és nemnegatív; ezt a továbbiakban feltesszük.

2.2. *A lokális determinisztikus modell*

Most a reakció mellett a diffúzióval végbemenő anyagtranszportot is figyelembe vesszük. Ha $\varphi=\varphi(r, t)$, $r\in\Omega\subset\mathbf{R}^1$, $t\in[0, \infty)$ a koncentráció, $D>0$ a diffúziós együttható, $\Delta=\frac{\partial^2}{\partial r^2}$ a *Laplace-operátor*, akkor

$$\frac{\partial\varphi}{\partial t}(r, t) = D\Delta\varphi(r, t)+f(\varphi(r, t)) \tag{3}$$

(a diffúzióval kísért reakció egyenlete, lásd pl. FIFE [6]). Az Ω tartomány lehet korlátos, vagy nem korlátos, és φ eleget kell, hogy tegyen perem- és kezdeti feltételeknek. A megoldások és invariáns halmazok létezését és egyértelműségét illetően KUIPER [8]-ra és AMMAN [1]-re utalunk, az aszimptotikus viselkedésről és a stacionárius állapotról lásd FIFE [5] és [6]-ot.

A továbbiak kedvéért kimondjuk a következő egzisztencia- és unicitási eredményt, amely KUIPER [8]-ból kiolvasható: Legyen $\Omega=(0,1)$; tegyük fel, hogy $c_d<0$, és válasszunk egy $\varrho>0$ számot, amely jobbra van $f(x)=0$ legnagyobb zérushelyétől. Tegyük fel, hogy a $\varphi(r, 0)=\varphi_0(r)$ kezdeti feltételre $0\leqq\varphi_0(r)\leqq\varrho$, $r\in[0, 1]$, $\varphi_0\in H_{bc}^2$ teljesül, ahol H_{bc}^2 $C^2[0, 1]$ azon függvényei halmazának $H^2[0, 1]$-beli[2] lezárását jelöli, amelyek teljesítik a peremfeltételeket. Legyenek a peremfeltételek

$$\varphi(0, t) = \varphi(1, t) = 0,$$

vagy

$$\partial\varphi/\partial r(0, t)-\alpha\varphi(0, t) = 0, \quad \partial\varphi/\partial r(1, t)+\beta\varphi(1, t) = 0 \quad (\alpha, \beta \geqq 0).$$

Ekkor létezik egyetlen olyan φ globális megoldása (3)-nak, amely kielégíti a kezdeti

[2] Lásd például GIHMAN, I. I. és SZKOROHOD, A. V. „*Bevezetés a sztochasztikus folyamatok elméletébe*" (Műszaki Könyvkiadó, Budapest, 1975.) 400. old. (A ford. megj.)

és peremfeltételeket, úgyhogy

$$\varphi \in C^1([0, \infty), L_2(0, 1)) \cap C([0, \infty), H^2(0, 1))$$

és

$$0 \leqq \varphi(r, t) \leqq \varrho$$

minden $r \in [0, 1]$ és minden $t \in [0, \infty)$ esetén.

2.3. *A globális sztochasztikus modell*

Most az X reaktáns részecskéinek t időpontbeli $X(t)$ számát tekintjük egy L hosszúságú térrészben. Ezt a függvényt egy ugró *Markov-folyamattal* jellemezzük, speciálisan egy **N** állapotterű születési-halálozási folyamattal, amelynek átmenetvalószínűségei:

$$P(X(t+s) = k+1 | X(s) = k) = p_{k,k+1}(t) = \lambda_k t + o(t), \qquad k \in \mathbf{N},$$

$$P(X(t+s) = k-1 | X(s) = k) = p_{k,k-1}(t) = \mu_k t + o(t), \quad k-1 \in \mathbf{N},$$

ahol a λ_k születési és a μ_k halálozási arányokat

$$\lambda_k = L\left(a_0 + a_1 \frac{k}{L} + a_2 \frac{k(k-1)}{2} + \ldots + a_p \frac{k(k-1)\ldots(k-p+1)}{L^p}\right) \tag{4}$$

és

$$\mu_k = L\left(b_1 \frac{k}{L} + b_2 \frac{k(k-1)}{L^2} + \ldots + b_q \frac{k(k-1)\ldots(k-q+1)}{L^q}\right)$$

adja meg. Bevezetve a

$$\lambda_L(x) = \sum_{i=0}^{p} a_i x\left(x - \frac{1}{L}\right)\ldots\left(x - \frac{i-1}{L}\right), \tag{5}$$

és

$$\mu_L(x) = \sum_{i=1}^{q} b_i x\left(x - \frac{1}{L}\right)\ldots\left(x - \frac{i-1}{L}\right)$$

függvényeket, azt írhatjuk, hogy

$$\lambda_k = L\lambda_L(k/L), \quad \mu_k = L\mu_L(k/L). \tag{6}$$

Vegyük észre, hogy $L \to \infty$ esetén minden x-re

$$\lambda_L(x) = \lambda(x) + O(1/L), \quad \mu_L(x) = \mu(x) + O(1/L).$$

A λ_k és μ_k átmeneti intenzitások egy kezdeti feltétellel együtt egyértelműen meghatározzák az $X(t)$ sztochasztikus folyamatot, feltéve vagy azt, hogy a reakció elsőrendű, vagy azt, hogy $c_d < 0$. Speciálisan a $p_k(t) = P(X(t) = k)$ valószínűségek egyértelmű megoldásai az úgynevezett *alapegyenletnek* (vagy a második *Kolmogorov-*

egyenletnek)[3]:

$$\dot{p}_k(t) = \lambda_{k-1}p_{k-1}(t)+\mu_{k+1}p_{k+1}(t)-(\lambda_k+\mu_k)p_k(t) \quad (k\in\mathbf{N})$$

a

$$p_k(0) = p_k^0, \quad \sum_{k\in\mathbf{N}} p_k^0 = 1$$

kezdeti feltételek mellett.

2.4. *A lokális sztochasztikus modell*

Most a teljes L hosszúságú térrészt N egyenlő, $l=L/N$ méretű cellára osztjuk fel, ahol az érintkező cellákat diffúzió köti össze, a cellákon belül pedig zajlik a reakció. Ha $X_j(t)$ jelöli az X reaktáns j-edik cellában levő részecskéinek számát a t időpontban, akkor az

$$X(t) = (X_1(t), \ldots, X_N(t))$$

függvényt egy $\mathbf{N}^N$ állapotterű ugró *Markov-folyamattal* modellezzük, amelynek átmeneti intenzitásai:

$$q_{k,k+e_j} = \lambda_{k_j}, \quad q_{k,k-e_j} = \mu_{k_j}, \quad j = 1, \ldots, N \quad \text{(reakció)} \tag{7}$$

$$q_{k,k+e_{j+1}-e_j} = (D^*/2)k_j, \quad j = 1, \ldots, N-1, \quad \text{(diffúzió)}$$

$$q_{k,k+e_{j-1}-e_j} = (D^*/2)k_j, \quad j = 2, \ldots, N,$$

$$q_{k,k+m} = 0,$$

egyébként. Itt $k=(k_1, \ldots, k_j, \ldots, k_n)$, $k+e_{j+1}-e_j$, $k+m\in\mathbf{N}^N$, e_j a j-edik egységvektor $\mathbf{R}^N$-ben, és

$$\lambda_{k_j} = l\lambda_l(k_j/l), \quad \mu_{k_j} = l\mu_l(k_j/l), \tag{8}$$

és $D^*>0$ a születési arány, a halálozási arány, illetve a diffúziós állandó, $\lambda_l(x)$ és $\mu_l(x)$ pedig az (5) által definiált függvények.

Peremfeltételeket is kell rögzítenünk. Ha semmi továbbit nem mondunk, akkor a peremen való visszaverődésre gondolunk (zéró fluxusú peremfeltételek). Ha a rendszer peremeinél előírt, rögzített (pl. nulla) koncentrációjú tartályokhoz van csatolva, ezt úgy modellezhetjük, hogy mindegyik peremhez egy cellát csatolunk rögzített számú részecskével, és az új cellát szomszédjához a fent megadott intenzitású diffúzióval csatoljuk.

A fent rögzített átmeneti intenzitások egy $\mathbf{N}^N$-en adott kezdeti eloszlással együtt egyértelműen meghatározzák az $X(t)$ sztochasztikus folyamatot, feltéve vagy azt, hogy a reakció elsőrendű, vagy hogy $c_d<0$. Speciálisan a $p_k(t)=P(X(t)=k)$ $k\in\mathbf{N}^N$, valószínűségek megoldásai a *többváltozós alapegyenletnek* (amely például

[3] Az egységes terminológia érdekében ajánlatos ezt magyarul *alap-*, vagy *állapotegyenletnek* (angolul — itt is —: *master equation*) nevezni, és az átmenetvalószínűségekre vonatkozókat első és második *Kolmogorov-egyenletnek* hívni. Az alapegyenlet a második *Kolmogorov-egyenlet* következménye. (A ford. megj.)

zéró fluxusú peremfeltételek esetén az alábbi alakú):

$$\dot{p}_k(t) = \sum_{j=1}^{N} (\lambda_{k_j-1} p_{k-e_j}(t) + \mu_{k_j+1} p_{k+e_j}(t) - (\lambda_{k_j} + \mu_{k_j}) p_k(t)) +$$

$$+ \frac{D^*}{2} \sum_{j=2}^{N} ((k_j+1) p_{k+e_j-e_{j-1}}(t) - k_j p_k(t)) +$$

$$+ \frac{D^*}{2} \sum_{j=1}^{N-1} ((k_j+1) p_{k+e_j-e_{j+1}}(t) - k_j p_k(t))$$

a $(p_k^0)_{k \in \mathbf{N}^N}$ kezdeti elosztással.

3. A modellek közötti kapcsolatok

Mivel a (globális vagy lokális) sztochasztikus modellt részletesebbnek tekintik a determinisztikusnál, az alapegyenletekből vissza kell, hogy tudjuk kapni a determinisztikus egyenleteket, ha az L térfogat végtelenhez tart (azaz termodinamikai határértékben). Másrészt, vissza kell, hogy tudjuk kapni a (sztochasztikus vagy determinisztikus) globális leírást a lokálisból, amikor a diffúzió dominál a reakcióhoz képest.

Ennek a szakasznak az a célja, hogy a fenti állításokat pontossá tegye és megmutassa, hogy milyen értelemben konzisztensek a modellek.

3.1. *Kapcsolat a globális sztochasztikus és determinisztikus modell között*

Ezt a kérdést teljesen megoldotta KURTZ [10, 11, 13] azáltal, hogy bebizonyította a következő eredményeket:

1. TÉTEL. A 2.1. és 2.3. szakaszban tett feltevések mellett fennállnak a következők:

(i) *A nagy számok törvénye:* Tegyük fel, hogy $\lim_{L\to\infty} X(0)/L = \varphi_0$ sztochasztikusan, akkor minden véges T-re és $\delta > 0$-ra

$$\lim_{L\to\infty} P\left(\sup_{0 \le t \le T} \left| \frac{X(t)}{L} - \varphi(t) \right| > \delta \right) = 0.$$

(ii) *A centrális határeloszlás-tétel:* Tegyük fel továbbá, hogy

$$\lim_{L\to\infty} \sqrt{L} \left(\frac{X(0)}{L} - \varphi_0 \right) = y_0$$

sztochasztikusan, akkor

$$\lim_{L\to\infty} \sqrt{L} \left(\frac{X(t)}{L} - \varphi(t) \right) = y(t)$$

gyengén (azaz a megfelelő valószínűségi mértékek konvergálnak gyengén[4]), ahol

[4] Lásd például GIHMAN és SZKOROHOD idézett könyvének 471. oldalát. (A ford. megj.)

$y(t)$ egy olyan *Gauss-típusú diffúziós folyamat*, amely megoldása a

$$dy(t) = \frac{d(\lambda-\mu)}{dx}(\varphi(t))y(t)dt + ((\lambda+\mu)(\varphi(t)))^{1/2}dw_t,\ y(0) = y_0$$

lineáris sztochasztikus differenciálegyenletnek.

(iii) *Diffúziós közelítés:* Van olyan valószínűségi mező, amelyiken

$$\sup_{0\leqq t\leqq T}\left|\frac{X(t)}{L} - u(t)\right| \leqq K_T\frac{\log L}{L},$$

ahol K_T egy T-től függő valószínűségi változó, és $u(t)$ egy olyan diffúziós folyamat, amelynek eloszlása azonos a

$$du(t) = (\lambda-\mu)(u(t))\,dt + \frac{1}{\sqrt{L}}((\lambda+\mu)(u(t)))^{1/2}dw_t,\ u(0) = X(0)/L$$

nemlineáris sztochasztikus differenciálegyenlet megoldásával.

Megjegyzés. Az 1. tétel (i) és (ii) állításában szereplő közelítések realizációkra is kimondhatók (legalábbis azon az áron, hogy elsőrendű reakciókra szorítkozunk, l. KURTZ [13]) a következő eredménnyel:

$$\begin{aligned} X(t)/L &= \varphi(t) + O(1/\sqrt{L}), \\ &= \varphi(t) + (1/\sqrt{L})y(t) + O(\log L/L), \\ &= u(t) + O(\log L/L), \end{aligned}$$

úgyhogy kiderül, hogy a *Gauss-folyamattal* és a diffúziós folyamattal való közelítés hibája azonos nagyságrendű.

3.2. *Kapcsolat a lokális sztochasztikus és determinisztikus modell között*

Amint kiderül, a diffúzió jelenléte arra kényszerít bennünket, hogy újraskálázzuk a hosszúságot, úgy hogy az új r tengelyen a rendszer a $[0, R]$ intervallumon legyen definiálva a

$$h = Rl/L = R/N$$

cellamérettel, ahol $h \to 0$ szükségszerűen. Attól függően, hogy a (3) egyenletet egy véges vagy egy végtelen intervallumon tekintjük: R=állandó, vagy $R \to \infty$. Az egyszerűség kedvéért feltesszük, hogy $R=1$, úgyhogy $h=l/L=1/N$.

Az $X(t)/l$ N-dimenziós vektorfolyamat egy $[0, 1]$-en definiált

$$x(r, t) = X_j(t)/l \quad \text{a} \quad ((j-1)/N, j/N] \quad \text{intervallumon}$$

lépcsős függvénynek felel meg. $x(., t) = x(t)$-t és (3) $\varphi(t)$ megoldását, mint $L_2(0, 1)$ elemeit fogjuk összehasonlítani, ahol az L, l és D^* paraméterek a sztochasztikus modellben úgy változnak, hogy $L \to \infty$ és $l/L = 1/N \to 0$. A (3)-beli D paramétert rögzítettnek tekintjük.

2. TÉTEL (ARNOLD és THEODOSOPOLU [3]). A 2.2. és 2.4. szakaszban tett feltevések mellett vegyünk állandó, vagy zéró fluxusú peremfeltételeket mind a két modellben. Tegyük fel, hogy $L\to\infty$, $N\to\infty$ úgy, hogy

(i) $\lim \|x(0)-\varphi_0\|_{L_2} = 0$,

(ii) $D^*/2 = DN^2$ (tehát $D^*\to\infty$),

(iii) $N^2/l = L^2/l^3 \to 0$ (tehát $l\to\infty$).

Ekkor fennáll *a nagy számok törvénye:* Minden véges T és $\delta>0$ esetén

$$\lim \sup_{0\leqq t\leqq T} P\big(\|x(t)-\varphi(t)\|_{L_2} > \delta\big) = 0.$$

Megjegyzés. A (ii) feltétel azért kell, hogy elbánjunk a *Laplace-operátorral.* A (iii) feltétel azt követeli meg, hogy az l cellaméret elég gyorsan tartson végtelenhez, például mint $l=L^{2/3+\varepsilon}$, $0<\varepsilon<1/3$. Másrészt, ha $L^2/l^3\to\infty$, akkor a nagy számok törvénye nem teljesül, úgyhogy (iii) nem hagyható el, ha L_2-normában való konvergenciát akarunk kapni. Ha azonban megelégszünk egy gyengébb fajta konvergenciával, azaz azzal, hogy

$$\int_0^1 \psi(r)x(r,t)\,dr \to \int_0^1 \psi(r)\varphi(r,t)\,dr$$

sztochasztikusan minden $\psi\in C^1[0,1]$ esetén, akkor (iii) elhagyható.

3. TÉTEL (ARNOLD és KOTELENEZ [2]). Tegyük fel, hogy a 2. tétel feltevései mellett még

$$\lim \big\|\sqrt{L}(x(0)-\varphi_0)-y_0\big\| = 0$$

sztochasztikusan is teljesül. Ekkor érvényes a következő *centrális határeloszlás-tétel:*

$$\lim \sqrt{L}(x(t)-\varphi(t)) = y(t) \quad \text{gyengén,}$$

(azaz a megfelelő valószínűségi mértékek konvergálnak gyengén a t időtől függő L_2-beli értékű függvényekre nézve), ahol $y(t)$ egy L_2-beli értékeket felvevő *Gauss-típusú diffúziós folyamat,* amely megoldása a

$$dy(t) = \frac{d}{dx}(\lambda-\mu)\big(\varphi(t)\big)y(t)\,dt + D\Delta y(t)\,dt + G\big(\varphi(t)\big)\,dw_t,\ y(0) = y_0$$

lineáris sztochasztikus parciális differenciálegyenletnek, ahol

$$T = GG^* = -D\frac{\partial}{\partial r}\varphi(t)\frac{\partial}{\partial r} + (\lambda+\mu)\big(\varphi(t)\big):\ H^1 \to H^{-1},$$

$$T \geqq 0,\quad G^*:\ H^1 \to L_2,\quad G:\ L_2 \to H^{-1},$$

és w_t az L_2-beli értékeket felvevő *Wiener-folyamat,* úgyhogy w_1 a *Wiener-mérték* a (H^1, L_2, i) absztrakt *Wiener-téren* a GROSS és KUO [9] által definiált értelemben.[5]

[5] (H, B, i) *absztrakt Wiener-tér,* ha H egy valós szeparábilis *Hilbert-tér, B* pedig az a *Banach-tér,* amelyet úgy kapunk, hogy H-t teljessé tesszük egy $\|\cdot\|_B$ mérhető normára nézve. $i: H\to B$ a kanonikus injekció. (A ford. megj.)

3.3. *Kapcsolat a lokális és a globális determinisztikus modell között*

Most (3) [0, 1]-en definiált $\varphi(r, t)$ megoldását tekintjük a 2.2.-ben megadott peremfeltételek mellett $D \to \infty$ esetén. Azt várjuk, hogy közvetlenül indulás után az erős diffúzió minden térbeli inhomogenitást kiegyenlít, úgyhogy $\varphi(r, t)$ közelítőleg megegyezik egy r-ben konstans $\psi(t)$-vel, ahol $\psi(t)$ megoldása a (2) globális kinetikai egyenletnek.

Először is jegyezzük meg, hogy a tekintett függvényosztály számára egyetlen olyan peremfeltétel van, amely egy nem nulla egyenesszakaszon teljesülhet: a zéró fluxusú peremfeltétel. Ennélfogva csak zéró fluxusú feltételek mellett van esély arra, hogy visszakapjuk a globális törvényt, ha „bekapcsoljuk" a diffúziót.

4. TÉTEL. Legyen $\varphi(r, t)$ (3) megoldása [0, 1]-en a

$$\frac{\partial \varphi(0, t)}{\partial r} = 0, \quad \frac{\partial \varphi(1, t)}{\partial r} = 0$$

zéró fluxusú peremfeltételek és a φ_0 kezdeti feltétel esetén. Legyen $\psi(t)$ (2) megoldása a

$$\psi_0 = \int_0^1 \varphi_0(r)\, dr$$

kezdeti feltétel mellett. Ekkor minden $\delta > 0$ és $T > 0$ esetén

$$\lim_{D \to \infty} \sup_{\delta \leq t \leq T} \|\varphi(r, t) - \psi(t)\|_{L_2} = 0.$$

Bizonyítás: A $D\Delta$ operátor L_2-n a

$$T(t)\, x(r) = c_0 + \frac{1}{\sqrt{2}} \sum_{n=1}^{\infty} c_n e^{-n^2 \pi^2 Dt} \cos n\pi r$$

által megadott $T(t)$ kontrakciós félcsoportot generálja, ahol

$$c_0 = \int_0^1 x(r)\, dr, \quad c_n = \frac{1}{\sqrt{2}} \int_0^1 x(r) \cos n\pi r\, dr, \quad x \in L_2$$

(lásd CURTAIN és PRITCHARD [4, 46. old.]). Minden térben állandó c-re $T(t)c = c$, és $\|T(t)\| = 1$. Továbbá $\lim_{D \to \infty} T(t) = \pi$ egyenletesen $\delta \leq t \leq T$ mellett, ahol π az 1 által generált egydimenziós altérre való vetítés, amit

$$\pi x(r) = \int_0^1 x(r)\, dr = c_0$$

definiál. Mindkét egyenletet félcsoport alakban felírva és figyelembe véve, hogy $\pi \varphi_0 = \psi_0$, kapjuk

$$\varphi(t) - \psi(t) = (T(t) - \pi)\varphi_0 + \int_0^t T(t-s)\big(f(\varphi(s)) - f(\psi(s))\big)\, ds.$$

Minden $[0, T]$ intervallumhoz és minden adott φ_0-hoz van olyan ϱ sugarú S_ϱ

gömb az altér-topológiában, amelyre $\varphi(r, t)\in S_\varrho$ és $\psi(t)\in S_\varrho$ minden $r\in[0, 1]$, $t\in[0, T]$ esetén. S_ϱ-ban f teljesíti a *Lipschitz-feltételt* az L_ϱ állandóval. Ennélfogva

$$\|\varphi(t)-\psi(t)\| = \|T(t)-\pi\|\|\varphi_0\| + L_\varrho \int_0^t \|\varphi(s)-\psi(s)\|\, ds.$$

Bevezetve az $\|T(t)-\pi\|\,\|\varphi_0\| = \varepsilon(t)$ jelölést a *Bellmann—Gronwall-lemma* szerint

$$\|\varphi(t)-\psi(t)\| \leqq \varepsilon(t) + L_\varrho \int_0^t \exp\left(L_\varrho(t-s)\right)\varepsilon(s)\, ds,$$

amiből az állítás következik.

3.4. *Kapcsolat a lokális és a globális sztochasztikus modell között*

Ismét zéró fluxusú peremfeltételeket feltételezünk a lokális sztochasztikus modellre (azaz visszaverődést a peremen) és azt az esetet tekintjük, amikor $D^*\to\infty$, L és l állandó. A részecskék

$$\bar{X}(t) = \sum_{j=1}^{N} X_j(t)$$

össz-száma nem *Markov-folyamat,* de azt várjuk, hogy nagy D^* esetén $\bar{X}(t)$ bizonyos értelemben közel van a globális sztochasztikus leíráshoz tartozó folyamathoz. Tettek már ilyen irányú kísérleteket, pl. MALEK-MANSOUR [14], de ismereteink szerint szigorú bizonyítást még nem adtak.

5. TÉTEL. Legyen $X(t)$ a (7)-ben és (8)-ban megadott átmeneti intenzitásokkal bíró lokális sztochasztikus modellhez tartozó sztochasztikus folyamat, legyen $\bar{X}(t)$ a részecskék össz-száma a lokális modellben[6]. Ekkor

$$\lim_{D^*\to\infty} \bar{X}(t) = Y(t) \quad \text{gyengén} \tag{9}$$

(azaz a megfelelő valószínűségi mértékek konvergálnak gyengén), ahol $Y(t)$ a (6)-ban megadott átmeneti intenzitásokkal bíró globális sztochasztikus modellhez tartozó, $\bar{X}(0)$ eloszlásával indított sztochasztikus folyamat.

Bizonyítás: Egy KURTZ [12] által kifejlesztett, félcsoport-approximációs módszert alkalmazunk. Az általánosság megszorítása nélkül feltehetjük, hogy folyamatunk leáll az

$$\mathbf{N}_\varrho^N = \left\{k\in\mathbf{N}^N:\ \bar{k} = \sum_1^N k_j \leqq \varrho\right\},$$

alakú halmazból való első kijutás időpontjában tetszőlegesen nagy ϱ esetén. Így egész vizsgálódásunkat erre a halmazra korlátozhatjuk.

Az $X(t)$ *Markov-folyamat* A_{D^*} infinitezimális generátorát az $\mathbf{N}_\varrho^N$-en értelmezett korlátos függvények $B(\mathbf{N}_\varrho^N)$ *Banach-terén*

$$A_{D^*} = D^* L_1 + L_2 .$$

[6] A felülvonás $\mathbf{N}^N$ elemeinél is a koordináták összegét fogja jelenteni. (A ford. megj.)

definiálja, ahol

$$L_1 f(k) = \frac{1}{2} \sum_{j=1}^{N-1} (f(k+e_{j+1}-e_j)-f(k))k_j + \frac{1}{2} \sum_{j=2}^{N} (f(k+e_{j-1}-e_j)-f(k))k_j$$

és

$$L_2 f(k) = \sum_{j=1}^{N} (f(k+e_j)-f(k))\lambda_{k_j} + (f(k-e_j)-f(k))\mu_{k_j};$$

itt λ_{k_j}-t és μ_{k_j}-t (8) adja meg, $k \in \mathbf{N}^N$. Az $Y(t)$ *Markov-folyamat* B infinitezimális generátorát az $\mathbf{N}_\varrho$-n értelmezett korlátos függvények $B(\mathbf{N}_\varrho)$ *Banach-terén*

$$Bf(j) = (f(j+1)-f(j))\lambda_j + (f(j-1)-f(j))\mu_j, \quad j \in \mathbf{N}_\varrho$$

definiálja, ahol λ_j-t és μ_j-t (6) adja meg.

Legyen M az összes olyan $f \in B(\mathbf{N}_\varrho)$ halmaza, amelyekhez létezik olyan $f_{D^*} \in B(\mathbf{N}_\varrho^N)$ sorozat, amellyel

$$\lim_{D^* \to \infty} \sup_k |f_{D^*}(k) - f(\bar{k})| = 0 \tag{10}$$

és

$$\lim_{D^* \to \infty} \sup_k |A_{D^*} f_{D^*}(k) - Bf(\bar{k})| = 0. \tag{11}$$

Be fogjuk bizonyítani, hogy $M = B(\mathbf{N}_\varrho)$. Mivel $\lambda - B$ valamilyen $\lambda > 0$-ra invertálható, KURTZ egy tétele [12, 630—631. old.] szerint ezen feltételek mellett (9) fennáll.

Úgy folytatjuk, miként PAPANICOLAU [16]; legyen minden $f \in B(\mathbf{N}_\varrho)$-ra

$$f_{D^*}(k) = f(\bar{k}) + \frac{1}{D^*} g(k), \quad k \in \mathbf{N}_\varrho^N,$$

ahol g-t még meg kell határoznunk. Mindenesetre (10) minden g-re érvényes. Továbbá

$$A_{D^*} f_{D^*}(k) = D^* L_1 f(\bar{k}) + L_1 g(k) + L_2 f(\bar{k}) + \frac{1}{D^*} L_2 g(k).$$

Mivel $\overline{k+e_{j+1}-e_j} = \bar{k}$, ezért $L_1 f(\bar{k}) = 0$. Így (11) teljesül, ha g megválasztható úgy, hogy

$$L_1 g(k) = -L_2 f(\bar{k}) + Bf(\bar{k}).$$

Figyeljük meg, hogy L_1 éppen annak a *Markov-folyamatnak* az infinitezimális generátora, amely $\mathbf{N}_\varrho^N$-en modellezi a tiszta, 1 intenzitású diffúziót visszaverődéssel a peremeken. Ez a folyamat ergodikus, P_0-lal jelölt stacionárius eloszlása a

$$p_k = \frac{m!}{k_1!\, k_2! \ldots k_N!} N^{-m}, \quad \text{ha} \quad \bar{k} = m, \quad k \in \mathbf{N}_\varrho^N,$$

polinomiális eloszlás, ahol $m \leqq \varrho$ a rendszerben kezdetben jelen levő részecskék száma (ami megmarad), és $p_k = 0$ egyébként (lásd VAN DEN BROECK, HORSTHEMKE és MALEK-MANSOUR [17]).

Ennélfogva $L_1 g = u$-nak van megoldása, feltéve, hogy az

$$\int u(x) P_0(dx) = \sum_k u(k) p_k = 0$$

megoldhatósági feltétel a mi speciális u-nkra teljesül. Ez az egyetlen dolog, amit még ellenőriznünk kell.

Fennáll, hogy

$$\sum_k (-L_2 f(k)+Bf(k))p_k = (f(m+1)-f(m))\left(\lambda_m - \sum_k \left(\sum_{j=1}^{N} \lambda_{k_j}\right) p_k\right) +$$

$$+(f(m-1)-f(m))\left(\mu_m - \sum_k \left(\sum_{j=1}^{N} \mu_{k_j}\right) p_k\right).$$

Az m részecske $N=L/l$ számú cellába való szétosztásakor kapott polinomiális eloszlásra

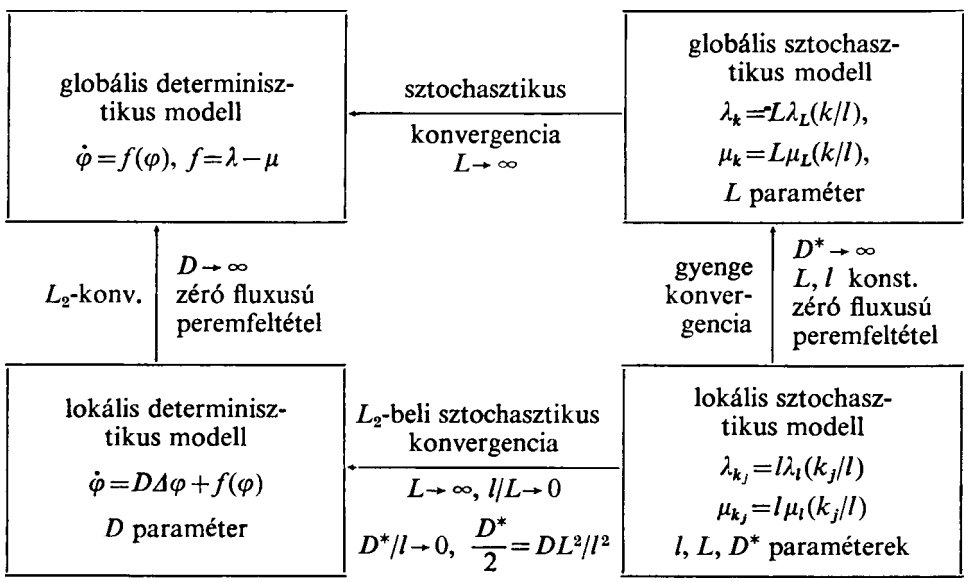
$$\sum_k \left(\sum_{j=1}^{N} \lambda_{k_j}\right) p_k = \sum_k \left(\sum_{j=1}^{N} l\lambda_l(k_j/l)\right) p_k =$$

$$= Nl \sum_{j=0}^{m} \lambda_l(k_j/l)\left(\frac{1}{N}\right)^j \left(1-\frac{1}{N}\right)^{m-j} \binom{m}{j} = L\lambda_L(m/L) = \lambda_m,$$

hasonlóan a μ-tagokra, úgyhogy a megoldhatósági feltétel valóban teljesül.

4. Összefoglalás

A négy modell közötti kapcsolatokat a következő sémán összegezzük:



IRODALOM

[1] AMANN, H., "Invariant sets and existence theorems for semilinear parabolic and elliptic systems," *J. Math. Anal. Appl.* **65** (1968) 432–467.
[2] ARNOLD, L. and KOTELENEZ, P., "Central limit theorem for the stochastic model of chemical reactions with diffusion" (in preparation).
[3] ARNOLD, L. and THEODOSOPOLU, M., "Deterministic limit of the stochastic model of chemical reactions with diffusion", *Adv. Appl. Prob.* **12** (1980) 367-379).
[4] CURTAIN, R. T. and PRITCHARD, A. J., *Infinite Dimensional Linear Systems Theory* Lecture Notes in Control and Information Sciences **8** (Springer, Berlin, Heidelberg, New York, 1978.)
[4] FIFE, P. C., "Asymptotic states for equations of reaction and diffusion", *Bull. Amer. Math. Soc.* **84** (1978) 693–726.
[6] FIFE, P. C., *Mathematical Aspects of Reacting and Diffusing Systems* Lecture Notes in Biomathematics **28** (Springer, Berlin, Heidelberg, New York, 1979.)
[7] HAKEN, H. *Synergetics* (Springer, Berlin, Heidelberg, New York, 1978.) (Second edition.)
[8] KUIPER, H. J., "Existence and comparison theorems for nonlinear diffusion systems", *J. Math. Anal. Appl.* **60** (1977) 166–181.
[9] KUO, H.-H., *Gaussian Measures in Banach Spaces* Lecture Notes in Mathematics **463** (Springer, Berlin, Heidelberg, New York, 1975).
[10] KURTZ, T., "Sclutions of ordinary differential equations as limits of pure jump Markov processes", *J. Appl. Prob.* **7** (1970) 49–58.
[11] KURTZ, T., "Limit theorem for sequences of jump Markov processes approximating ordinary differential processes", *J. Appl. Prob.* **8** (1971) 344–356.
[12] KURTZ, T., "Semigroups of conditioned shifts and approximation of Markov processes", *Annals of Prob.* **3** (1975) 618–642.
[13] KURTZ, T., "Strong approximation theorems for density dependent Markov chains", *Stoch. Proc. and their Appl.* **6** (1978) 223–240.
[14] MALEK-MANSOUR, M., "*Fluctuation et transition de Phase de Non-equilibre dans les Systems Chimiques*", These (Université Libre de Bruxelles, 1979).
[15] NICOLIS, G. and PRIGOGINE, I., *Selforganization in Non-equilibrium Systems* (Wiley, New York, 1977).
[16] PAPANICOLAU, G. C., "Asymptotic analysis of stochastic equations" in: *Studies in Probability Theory* Vol. **18**. Ed. M. Rosenblatt (The Mathematical Association of America, 1978).
[17] VAN DEN BROECK, C., HORSTHEMKE, W. and MALEK-MANSOUR, M., "On the diffusion operator of the multivariate master equation", *Physica* **89**A (1977) 339–352.

Jegyzet: Miután a dolgozat elkészült, H. G. OTHMER felhívta figyelmünket a CONWAY, E., HOFF, D. and SMOLLER, J., "Large time behaviour of systems of nonlinear reaction-diffusion equations", *SIAM J. Appl. Math.* **35** (1978) 1–16. dolgozatra, amelyben a 4. tételhez hasonló eredményeket bizonyítottak be.

FORDÍTOTTA:
TÓTH JÁNOS
MTA SZTAKI
1111 BUDAPEST, KENDE U. 13—17.

A kiadásért felel az Akadémiai Kiadó igazgatója
Műszaki szerkesztő: Sándor István
A kézirat nyomdába érkezett: 1982. II. 5. — Terjedelem: 14,70 (A/5) ív
82-54 — 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

VII. KÖTET

AKADÉMIAI KIADÓ, BUDAPEST
1981

TARTALOMJEGYZÉK

INDEX

82-54 — Szegedi Nyomda — Felelős vezető: Dobó József igazgató