

# SZOFTVER

MIHÁLYFFY LÁSZLÓ — BAGDY GÁBOR

## Nagyméretű dinamikus nemlineáris ökonometriai modellek szabályozására alkalmas számítógépi programok

### Bevezetés

A gazdaság dinamikájával és szabályozásának lehetőségeivel foglalkozó kutatások rendkívül széleskörűek; megállapítható, hogy e tudományág fejlesztéséből a magyar kutatók is kivették a részüket. Az ide tartozó tanulmányok, publikációk jelentős hányada elméleti jellegű [17], [19] stb. — sok esetben találkozunk pl. a gazdaság lehetséges szabályozó változóira vonatkozó kvalitatív megállapításokkal, melyeket a gazdasággal kapcsolatos bizonyos matematikailag megfogalmazható feltételekből lehet levezetni —, nem hiányoznak azonban azok a munkák sem, amelyekben az elméleti megfontolások alapján kialakított dinamikus modellt és vele együtt a bizonyos kritériumok szerint optimálisnak tekinthető szabályozást konkrétan számszerűsítik. Az utóbbi esetben, pontosabban, az ilyen esetek többségében általában — mind itthon, mind pedig határainkon kívül — az a jellemző, hogy a számszerűsített modellek általában kisméretűek [18], változóik erősen aggregált mennyiségekre, mutatókra vonatkoznak, s így az eredmények a felhasználók számára csupán orientáló információt jelentenek. Nyilvánvaló, hogy a gazdaság tervezői számára hasznos lenne, ha adott gazdaságpolitikai célok és adott gazdasági szabályozók mellett minél részletesebb előzetes információhoz lehetne jutni a különböző gazdaságpolitikai döntések várható következményeivel kapcsolatban; ez nagyobb méretű modellek kialakítását, számszerűsítését és szabályozását teszi szükségessé, mint amilyenekkel az optimális szabályozással foglalkozó irodalomban korábban találkozhattunk.

A nagy, vagy akár csak viszonylag nagyobb méretű dinamikus modellek létrehozásával kapcsolatos elvi és gyakorlati nehézségek széles körben ismeretek. Anélkül, hogy az ún. „nagy modellekkel” kapcsolatos számos fenntartás és kritikai észrevétel jogosságát megkérdőjeleznénk, úgy gondoljuk, hogy bizonyos esetekben érdemes engedményeket tenni az egzakttság rovására, ha történetesen olyan modellel van dolgunk, amely, ha elvileg nem is minden vonatkozásban kifogástalan, bizonyos evidenciák (pl. ex post előrejelzések) tanúsága szerint egyes jelenségeket, folyamatokat mégis helyesen ábrázol, s ily módon megfelelő keretek között értelmes következtetések levonását teszi lehetővé. Más szóval, abból indulunk ki, hogy vannak, és pedig nem is kizárólag csak l'art pour l'art szempontoknak eleget tevő, nagy vagy inkább közepes méretű, számszerűen meghatározott dinamikus ökonometriai modellek, amelyek bizonyos szabályozó funkciójú változók (pl. támogatások, adókulcsok stb.) hatásának vizsgálatára alkalmasak, ha csak rendelkezésünkre állnak az ehhez szükséges számítástechnikai eszközök, a bizonyos szempontok szerint optimális

szabályozást meghatározó gépi programok. Ebben a dolgozatban két olyan programcsomagot mutatunk be, amely ennek a feladatnak a megoldására, tehát nagy vagy inkább közepes méretű ökonometriai modell szabályozására alkalmas: az egyik GREGORY C. CHOW [5], [6], a másik ARNE DRUD [7], [8] nevéhez fűződik. (A méretekkal kapcsolatban mondottakat itt még azzal egészítjük ki, hogy szabályozás szempontjából gyakran nagyon bizonyul az a modell, amely becslési szempontból közepes méretű.) Ez idő szerint már mindkét programcsomag Magyarországon is a felhasználók rendelkezésére áll.<sup>1</sup> A két program két, lényegesen különböző megközelítést reprezentál: ezeket külön-külön fogjuk ismertetni, majd az eddigi hazai számítástechnikai tapasztalatokat a két program működését összehasonlítva, együtt fogjuk tárgyalni.

A Chow- és a Drud-féle programcsomag — ez utóbbinak a neve CONOPT (= *Control Optimal*) — már a megoldandó feladatok definíciójában is eltér egymástól, ezeket — ti. a megoldható feladatok típusát — azonban célszerű egyidejűleg tárgyalni, mivel így mind a közös, mind pedig az eltérő vonások jól érzékelhetők. Induljunk ki azokból a dinamikus nemlineáris ökonometriai modellekből, amelyek az alábbi alakba írhatók:

$$(0.1) \quad g(y_t, y_{t-1}, \dots, y_{t-\tau}, x_t, x_{t-1}, \dots, x_{t-\theta}) = \varepsilon_t, \\ t = 1, 2, \dots, T,$$

ahol

$t$  jelöli az időt,  $t = 1, 2, \dots, T$ , és  $T$  a vizsgált időszak hossza (időhorizont);

$y_t = (y_{1,t}, y_{2,t}, \dots, y_{n,t})'$  a modell *endogén* változóinak  $n$ -dimenziós vektora a  $t$  időszakban (a vessző transzponálást jelöl);

$x_t = (x_{1,t}, x_{2,t}, \dots, x_{m,t})'$  a modell *szabályozó* változóinak  $m$ -dimenziós vektora a  $t$  időszakban;

$\tau$  illetve  $\theta$  a késleltetés maximális mértékét jelöli a modellben az endogén, ill. a szabályozó változókra vonatkozóan;

$\varepsilon_t = (\varepsilon_{1,t}, \varepsilon_{2,t}, \dots, \varepsilon_{n,t})'$  a véletlen zavarások nulla várható értékű vektora a  $t$  időszakban. Fel fogjuk tenni, hogy a  $V = (\text{cov}(\varepsilon_{i,t}, \varepsilon_{j,s}))$  kovariancia-mátrix nullával egyenlő, hacsak  $t \neq s$ ;  $t = s$  esetén  $V$  értéke nem függ az időtől, és azt is feltesszük, hogy rendelkezésünkre áll  $V$  valamilyen becslése;

$g(\cdot) = (g_1(\cdot), g_2(\cdot), \dots, g_n(\cdot))'$  folytonos leképezés  $R^{(\tau+1)n + \theta + 1}m$ -ből  $R^n$ -be, kétszer folytonosan differenciálható komponensekkel.

Az eddig elmondottakat a következőkkel kell kiegészítenünk. (0.1) skalár alakba — tehát  $g_1, g_2, \dots, g_n$  segítségével — átírva, nyilvánvalóan fölfogható, mint egy dinamikus nemlineáris ökonometriai modell szimultán egyenleteinek rendszere, egyszerű átrendezéstől eltekintve strukturális formában. A  $g$  leképezésben — bár ezt a jelölés nem tükrözi — bizonyos becslt, tehát adottnak tekinthető együtthatók, paraméterek függhetnek a  $t$  időtől. Ami a modell változóinak endogén és szabályozó változókra való felosztását illeti, ez szokatlannak tűnhet, mivel az ökonometriában a modellek változóit endogén és predeterminált változókra, és ezen belül az utóbbiakat késleltetett endogén és exogén változókra szokás felosztani. A kétféle terminológia között a kapcsolat a következő. A szabályozás szempontjából az endogén változóknak a vizsgált időszakhoz

<sup>1</sup> A programokat az MTA Számítástechnikai és Automatizálási Kutató Intézet IBM 3031 számítógépén installáltuk.

( $t = 1, 2, \dots, T$ ) tartozó összes értékét változónak (ismeretlennek) kell tekinteni, és ezen kívül az exogén változók közül ki kell választani azokat, amelyeknek az értékét befolyásolni lehet. Ez utóbbiak, esetleges késleltetéseikkel együtt, lesznek a szabályozó változók, míg az összes többi exogén változó  $t = 1, 2, \dots, T$  időszakhoz tartozó értéke a szabályozás szempontjából adott-ság, paraméter lesz. *Fontos követelmény, hogy mind az endogén, mind pedig a szabályozó változóknak a  $t = 1$  időszakot megelőző értékeit ( $y_0, y_{-1}, \dots, y_{1-T}, x_0, x_{-1}, \dots, x_{1-t}$ ) ismernünk kell, egyébként a feladat határozatlan lenne.*

Rendeljünk mármost minden endogén és szabályozó változóhoz egy alsó és egy felső korlátot a következőképpen:

$$(0.2) \quad \underline{y}_t \leq y_t \leq \bar{y}_t, \quad (t = 1, 2, \dots, T)$$

$$(0.3) \quad \underline{x}_t \leq x_t \leq \bar{x}_t.$$

Nem zárjuk ki azt a lehetőséget, hogy bizonyos alsó határok mínusz végtelennel, bizonyos felső határok pedig plusz végtelennel legyenek egyenlők, nyilvánvaló viszont, hogy a változók egy részének értéke csak akkor lehet értelmes, ha bizonyos véges határok közé esik.

*Megengedett szabályozásnak* v. *vezérlésnek* nevezve mármost a szabályozó változók értékének bármely, a (0.3)-at kielégítő rendszerét, fel fogjuk tenni, hogy (0.1)-nek az endogén változókra vonatkozóan minden megengedett szabályozás mellett van megoldása.

A (0.1) modellből és a (0.2)–(0.3) feltételekből úgy jutunk optimális szabályozási feladathoz, hogy kijelölünk valamilyen gazdaságpolitikai célt, amely kifejezhető, mint az endogén és a szabályozó változóknak valamilyen valós skalár értékű függvénye a vizsgált időszakban. Legyen ez a függvény  $f(y_1, y_2, \dots, y_T, x_1, x_2, \dots, x_T)$ ; a gazdaságpolitikai optimalitás ennek valamilyen szélsőértékével kapcsolatos, tegyük fel, hogy minimumról van szó. Mivel  $y_1, \dots, y_T, x_1, \dots, x_T$  az  $\varepsilon_t$  véletlen zavaráson keresztül a véletlentől is függ,  $f$  értéke is valószínűségi változó, és ezért a várható értékének a minimalizálására kell törekednünk:

$$(0.4) \quad M(f(y_1, y_2, \dots, y_T, x_1, x_2, \dots, x_T)) = \min!$$

Technikai szempontból az  $f$  célfüggvényről ugyanazt tesszük fel, mint a (0.1) modell egyenleteiben szereplő függvényekről, azaz, kétszeri folytonos differenciálhatóságot. Ettől eltekintve a célfüggvény alakjára vonatkozóan általában nincs semmi megszorítás, és így igen sokféle gazdasági célkitűzés megfogalmazására van lehetőség (külkereskedelmi mérleghiány minimalizálása a vizsgált időszakra vonatkozóan, az import minimalizálása bizonyos relációban stb.). Széles körben használják az ún. kvadratikus célfüggvényt, amelynél az endogén változókra vonatkozóan<sup>2</sup> előírnak egy optimálisnak tekintett

$$\hat{y}_1, \hat{y}_2, \dots, \hat{y}_T$$

értékek által meghatározott pályát, és a követelmény ebben az esetben az, hogy a modell változói ezeket a célértékeket minél jobban megközelítsék. A

<sup>2</sup> Bizonyos esetekben technikai okokból a szabályozó változókhoz — vagy ezek egy részéhez — is rendelnek célkitűzéseket.

közelítés mértékét a célfüggvény méri, amely ekkor — a várható érték operátorától eltekintve —

$$(0.4a) \quad f = \sum_{t=1}^T (y_t - \hat{y}_t)' K_t (y_t - \hat{y}_t)$$

alakú, ahol  $t = 1, 2, \dots, T$  esetén  $K_t$  adott pozitív szemidefinit mátrix. Az a körülmény, hogy a  $K_t$  mátrixoknak nem kell pozitív *definitnek* lenniök, azt jelenti, hogy nem szükséges minden egyes skalár értelemben vett endogén változóhoz célkitűzést rendelnünk. A szabályozási feladat meghatározásának teljessé tételéhez most már csak azt kell megjegyeznünk, hogy  $y_t$  és  $x_t$  ( $t = 1, 2, \dots, T$ ) valószínűségi változó jellege miatt a (0.2)–(0.3) feltételek teljesülését nem vehetjük biztosra, és így ezeket a

$$(0.2a) \quad P(\underline{y}_t \leq y_t \leq \bar{y}_t) > p_t$$

$$(0.3a) \quad P(\underline{x}_t \leq x_t \leq \bar{x}_t) > q_t$$

követelményekkel kell helyettesítenünk, ahol  $p_1, p_2, \dots, p_T, q_1, q_2, \dots, q_T$  alkalmasan választott (egynél kisebb) valószínűségek. Ily módon tehát a következő feladathoz jutottunk:

*minimalizáljuk az  $f(y_1, \dots, y_T, x_1, \dots, x_T)$  célfüggvény várható értékét (lásd (0.4)) a (0.1) modell valamint a (0.2a)–(0.3a) feltételek mellett.*

Ebből a *sztochasztikus programozási feladathoz* bizonyos, és pedig lényeges egyszerűsítések révén jönnek létre azok a feladatok, amelyeket a Chow-féle programmal, illetve a CONOPT segítségével meg lehet oldani. G. C. CHOW felfogása szerint ([5], 161–163. oldal) az ökonometriai modellek szabályozásánál a véletlen szerepének figyelembe vétele sokkal fontosabb, mint a változók (0.2)–(0.3), illetve (0.2a)–(0.3a) típusú feltételekkel való korlátozása, e feltételek ugyanis helyes modell-specifikáció esetén eo ipso teljesülnek, kivált ha teljesen általános alakú célfüggvények helyett (0.4a) alakú kvadratikusságú célfüggvényekre szorítkozunk. Ilyen típusú célfüggvények esetén a  $K_t$  mátrixok alkalmas választásával valóban befolyásolható a változók értékének alakulása, és így az irreális értékek kiküszöbölése; szükség esetén a szabályozó változókat is be lehet venni a célfüggvénybe. Ennek a megközelítésnek a kritikájára később még visszatérünk, egyelőre azt szögezzük le, hogy a Chow-féle program, illetve algoritmus esetén egy (0.4a) típusú függvény várható értékének a minimalizálása a feladat a (0.1) modell mint korlátozó feltétel mellett.

A. DRUD, a CONOPT programcsomag tervezője más megoldást választott. Ő a véletlen hatások vizsgálatáról mondott le, és ily módon egy determinisztikus feladathoz jutott, melynek célfüggvénye  $f(y_1, \dots, y_T, x_1, \dots, x_T)$ , feltételrendszere egyrészt (0.2)–(0.3)-ból, másrészt pedig a (0.1) modelltől áll, ennél azonban a jobb oldalon álló  $\varepsilon_t$  véletlen zavarást most 0-val kell helyettesíteni. A következőkben először a Chow-féle algoritmus, azután pedig a CONOPT főbb sajátosságait fogjuk ismertetni. Megjegyezzük, hogy mindkét program szempontjából a modell egyenletei a bennük szereplő paraméterekkel együtt a bemenő adatokhoz tartoznak, azokat tehát valamilyen további program segítségével kell becsülni.



# I. Kvadratikus célfüggvény várható értékének minimalizálása nemlineáris sztochasztikus modell esetén (G. C. Chow algoritmus)

Az előzőekben alkalmazott jelölések kismérvű módosításával a vizsgált feladat az alábbi alakba írható:

*minimalizáljuk az*

$$(1.1) \quad M \sum_{t=1}^T (y_t - \hat{y}_t)' K_t (y_t - \hat{y}_t)$$

*kifejezést az*

$$(1.2) \quad y_t = g(y_t, y_{t-1}, x_t, w_t) + \varepsilon_t, \\ t = 1, 2, \dots, T$$

*modell mint feltételrendszer mellett.*

Mielőtt tisztáznánk (1.2) és a bevezetésben felírt (0.1) modell kapcsolatát, néhány általános jellegű észrevételt kell elmondanunk a Chow-féle szabályozási koncepcióval kapcsolatban. Az (1.1) célfüggvényt, amely a gyakorlatban sok esetben súlyozott négyzetösszegre redukálódik, gyakran kifogásolják — többek között hazai kutatók is — mondván, hogy az szimmetrikus jellege miatt nem tesz különbséget a kitűzött céloktól felfelé vagy lefelé való eltérés között, holott nyilván nem mindegy az „összjólét” szempontjából, hogy pl. a nem rubel elszámolású külkereskedelmi mérleg aktívuma a tervezett 400 millió dollár helyett 200 vagy 600 millió lesz. Ezen kívül sokszor megemlítik, hogy (1.1) mint a gazdasági tervezés segédeszköze, igen sok szubjektív elemet tartalmaz. Ez a megállapítás elsősorban a súlyok megválasztásának kérdésére vonatkozik, és lényegében nem vitatható. Tovább rontja a helyzetet az, ha a súlyoknak eredeti feladatuk mellett még azt a szerepet is be kell tölteniük, hogy megakadályozzák irreális értékek létrejöttét a szabályozás során, vagyis, amikor a súlyoknak a (0.2a)–(0.3a) egyenlőtlenségeket is helyettesíteniük kell. Ezekkel a jogos kifogásokkal szemben azt kell megemlíteni, hogy az Egyesült Államokban és Nyugat-Európában számos esetben alkalmazták a Chow-féle szabályozási programot, és általában eredményesen. A sikeres alkalmazások példaként megemlítjük

- a Michigan-i Egyetem negyedéves ökonometriai modelljét [6],
- a SOVMOD modellcsalád egyes változatait<sup>3</sup> [14],
- J. BRADA, A. KING és D. SCHLAGENHAUF (Egyesült Államok) csehszlovákiai modelljét [2],
- a Rheinisch-Westfälisches Institut für Wirtschaftsforschung (NSZK) negyedéves konjunktúra-modelljét [10],

<sup>3</sup> Az Egyesült Államokban a hetvenes években dolgozták ki Donald Green professzor vezetésével az ún. SOVMOD (= *Soviet Model*) modelleket, elsősorban a szovjet–amerikai külkereskedelem alakulásának prognosztizálására. Green véleménye szerint az ex post jellegű vizsgálatokban a megfelelően választott súlyok többek között pl. egyértelműen tükrözték a különbséget N. Sz. Hruscov és L. I. Brezsnjev személyisége között; ezt a következtetést azonban célszerű bizonyos óvatossággal fogadnunk.

- az NSZK gazdaságát leíró, Keynes-i alapokon nyugvó ökonometriai modellt [12],
- az osztrák OPTPOL-1 modellt [20] stb.

E tapasztalatok fényében célszerűtlen lenne a Chow-féle megközelítést heurisztikus elemeire való hivatkozással elvetni; közismert, hogy az alkalmazott matematika számos területén használunk részben vagy egészen heurisztikus alapon működő módszereket.

Áttérve mármost az (1.2), illetve a korábban felírt (0.1) modell közötti formális különbségekre, azt látjuk, hogy (1.2)-ben

- az endogén változók egynél magasabb rendű késleltetett értékei nem szerepelnek a modellben;
- a szabályozó változók késleltetett értékei nem fordulnak elő;
- szerepel egy  $w_t$  paraméter.

Ez utóbbi most — mint vektor — azoknak az exogén változóknak az összességét jelöli, amelyek a modell becslésénél szerepeltek, szabályozó funkciójuk azonban nincsen. Könnyen belátható, hogy amennyiben egy modellben az  $y_{t-2}, y_{t-3}, \dots, x_{t-1}, x_{t-2}, \dots$  stb. változók is szerepelnek, akkor azokat egyszerű *azonosságok* segítségével ki lehet küszöbölni — pl.  $y_{1, t-2}$  helyett  $y_{n+1, t-1}$ -et írunk, ha az endogén változók vektora ez ideig  $n$ -dimenziós volt, s. í. t. — s így módon minden dinamikus modell az (1.2) alakra hozható. Amennyiben ilyen kiküszöbölésre szükség van, akkor ezekhez — mint a modell azonosságaihoz — értelemszerűen nem tartozik véletlen zavarás, vagyis ilyen esetben  $\varepsilon_t$  bizonyos komponensei és ennek megfelelően a kovariancia-mátrix bizonyos sorai, ill. oszlopai nullával lesznek egyenlők. A modellben egyébként természetesen szerepelhetnek a kiküszöbölés céljára bevezetett azonosságok mellett egyéb azonosságok is. Az a körülmény, hogy (1.2)-ben a bal oldal az egyes endogén változók kifejezését adja meg, a strukturális forma eredetére (a becslés folyamatára) utal.

G. C. Chow-nak az (1.1)–(1.2) nemlineáris szabályozási feladatra adott eljárása egy lineáris modell szabályozására kidolgozott algoritmus általánosítása. Célszerű ezt az eredeti — egyébként ugyancsak Chow nevéhez fűződő — eljárást bemutatni, éspedig nemesupán azért, mert így a nemlineáris módszer leírását is áttekinthetőbbé lehet tenni, hanem azért is, mivel az eredeti eljárás minden tekintetben korrekt, míg a nemlineáris feladatra adott eljárás egy sor heurisztikus elemet is tartalmaz. Tekintsük tehát azt az esetet, amikor (1.2) szerepét a következő lineáris modell veszi át:

$$(1.3) \quad y_t = A_t y_{t-1} + C_t x_t + b_t + \varepsilon_t, \\ t = 1, 2, \dots, T;$$

itt  $A_t$  és  $C_t$  adott mátrixok,  $b_t$  pedig adott vektor, amely a nem szabályozó funkciójú exogén változók valamilyen lineáris kifejezése — az (1.2)-ben szereplő  $w_t$  megfelelője —; az a körülmény, hogy (1.2)-vel ellentétben  $y_t$  most nem szerepel a jobb oldalon, esetleges átrendezést jelent a strukturális formához képest.

A Chow által javasolt megoldás az ún. Bellman-elven alapul. Ez azt mondja ki, hogy amennyiben adott a  $t = 1, 2, \dots, T$  időhorizonton az (1.1), (1.3)

feladatnak egy optimális

$$x_1, y_1, x_2, y_2, \dots, x_T, y_T$$

megoldása, akkor ez optimális az időhorizont bármely részére vonatkozóan is. Ebből az következik, hogy a  $T$  időszakra vonatkozóan az  $x_T$  optimális vezérlés és az ehhez tartozó  $y_T$  trajektória-elem meghatározható mint  $x_1, y_1, \dots, x_{T-1}, y_{T-1}$  függvénye, hiszen  $y_T$ -től az (1.1) célfüggvénynek csupán az utolsó tagja függ. Ezzel kapcsolatban a következő segédtételekre lesz szükségünk:

1. lemma. Legyen

$$(1.4) \quad y = Ay_{-1} + Cx + b + \varepsilon$$

az (1.3) modellnek valamelyik  $1 \leq t \leq T$  időszakhoz tartozó egyenlete; itt tehát — későbbiekben nyilvánvalóvá váló szempontok miatt — elhagytuk a  $t$  indexet, és  $y_{t-1}$  helyett  $y_{-1}$ -et írtunk. Legyen továbbá  $\hat{y}$  tetszőleges,  $y$ -nal megegyező dimenziójú konstans vektor,  $K$  pedig olyan, az  $y$ -nak megfelelő méretű pozitív szemidefinit mátrix, amelyre  $C'KC$  nemszinguláris. Állítás: tetszőleges  $y_{-1}$  esetén az

$$(1.5) \quad M(y - \hat{y})' K(y - \hat{y})$$

kifejezés az (1.4) feltétel mellett akkor és csak akkor lesz minimális, ha

$$(1.6) \quad x = Gy_{-1} + q,$$

ahol

$$(1.7) \quad G = -(C'KC)^{-1}C'KA, \quad q = (C'KC)^{-1}C'K(\hat{y} - b - \varepsilon).$$

*Bizonyítás.*<sup>4</sup> Mivel  $K$  pozitív szemidefinit, van olyan pozitív szemidefinit  $L$  mátrix, amelyre  $K = L'L$ .

Ennek következtében a

$$z = (y - \hat{y})' K(y - \hat{y})$$

kifejezést  $z = r'r$  alakba írhatjuk, ahol

$$r = LAy_{-1} + LCx + Lb + L\varepsilon - L\hat{y};$$

itt  $y$ -nak (1.4) alatti előállítását is figyelembe vettük.

Legyen

$$Q = LC(C'KC)^{-1}C'L';$$

$L'L = K$  miatt ez éppen az  $LC$  mátrix képterére vonatkozó szimmetrikus projektor. Mivel nyilván  $r = Qr + (E - Q)r$ , ahol  $E$  jelöli a megfelelő méretű egységmátrixot, a  $z$  kifejezést a következő alakba is írhatjuk:

$$z = r'Qr + r'(E - Q)r.$$

<sup>4</sup> Mivel dolgozatunk ismertető jellegű, bizonyításokat csak elvéve közlünk, és csak olyan esetekben, amikor ezt valami külön szempont indokolja. A jelen esetben azért került erre sor, mivel G. C. Chow bizonyítása pontatlan.

Itt most a második tag a jobb oldalon egyrészt nemnegatív, másrészt

$$(E - Q)LC = 0$$

miatt  $x$  értékétől független. Ily módon, mivel  $r'Qr$  is nemnegatív,

$$z \geq r'(E - Q)r,$$

és itt egyenlőségjel akkor és csak akkor fog állni, ha  $r'Qr = 0$ , vagyis, ha

$$Qr = QLAy_{-1} + QLCx + QL(b + \varepsilon - \hat{y}) = 0.$$

Ezt az egyenletet balról  $C'L'$ -vel szorozva, a vele ekvivalens

$$C'KAy_{-1} + C'KCx + C'K(b + \varepsilon - \hat{y}) = 0$$

egyenlethez jutunk, amely, mint könnyen belátható, éppen akkor teljesül, ha  $x$ -et (1.6) és (1.7) határozza meg. Ez tehát az az  $x$  vektor, amelyre a  $z$  kifejezés tetszőleges  $y_{-1}$  mellett minimális lesz, és a várható érték monotonitása miatt ugyanerre az  $x$ -re veszi fel (1.5) is a minimumát. Konvex célfüggvényről lévén szó, megfontolásainkból egyidejűleg következik a feltételeknek mind a szűkségessége, mind az elegendősége.

A Bellman-elv alapján mármost ahhoz, hogy az  $x_t$  szabályozás a  $t = 1, 2, \dots, T$  időhorizonton optimális legyen,  $t = T$ -re teljesülniök kell a következő összefüggéseknek:

$$(1.8) \quad x_T = G_T y_{T-1} + q_T$$

$$(1.9) \quad G_T = -(C'_T K_T C_T)^{-1} C'_T K_T A_T, \quad q_T = (C'_T K_T C_T)^{-1} C'_T K_T (\hat{y}_T - b_T - \varepsilon_T).$$

A lényeg itt az, hogy  $G_T$  és  $q_T$  értékét  $x_T$  és  $y_{T-1}$  értékétől függetlenül meg lehetett határozni. (1.8)-nak  $t$  tetszőleges értékéhez tartozó

$$(1.10) \quad x_t = G_t y_{t-1} + q_t$$

általánosítását *lineáris visszacsatolásnak* nevezzük. Nyilvánvaló, hogy amennyiben  $G_t$  és  $q_t$  minden  $t$  időszakra ismert, akkor az endogén és a szabályozó változók értéke az (1.3) modell alapján  $t = 1$ -ből kiindulva, szukcesszive meghatározható. Könnyen belátható a következő segédétel helyessége:

**2. lemma.** Ha az (1.3) modell esetén minden időszakra fennáll (1.10), továbbá  $G_t$  minden véletlen hatástól független, míg  $q_t$  csupán a  $t$ -edik időszakhoz tartozó véletlen zavarástól,  $\varepsilon_t$ -től függ, akkor  $y_t$  csak  $\varepsilon_1$ -től,  $\varepsilon_2$ -től,  $\dots$ ,  $\varepsilon_t$ -től függ. Ebből következően  $y_{t-1}$  és  $\varepsilon_t$  statisztikailag független egymástól.

Az (1.3) modell valamint (1.8)–(1.9) segítségével  $y_T$  az (1.1) célfüggvényből kiküszöbölhető úgy, hogy az formálisan már csak  $y_1$ -től,  $y_2$ -től,  $\dots$ ,  $y_{T-1}$ -től fog függni. Ehhez azt kell megjegyezni, hogy

– egyrészt a 2. lemma miatt

$$M \varepsilon'_T K_T (A_T + C_T G_T) y_{T-1} = 0,$$

– másrészt az 1. lemma bizonyításából következően

$$q'_T C'_T K_T (A_T + C_T G_T) = 0.$$

Ennek, valamint az előbb mondottaknak figyelembe vételével (1.1) utolsó két tagja így írható:

$$(1.11) \quad M(y_{T-1} - \hat{y}_{T-1})' K_{T-1}(y_{T-1} - \hat{y}_{T-1}) + \\ M[(A_T + C_T G_T)y_{T-1} + C_T q_T + b_T + \varepsilon_T - \hat{y}_T]' K_T [(A_T + C_T G_T)y_{T-1} + \\ + C_T q_T + b_T + \varepsilon_T - \hat{y}_T] = M\hat{y}'_{T-1} [K_{T-1} + (A_T + C_T G_T)' K_T (A_T + \\ + C_T G_T)] y_{T-1} - 2M[\hat{y}'_{T-1} K_{T-1} + (\hat{y}_T - b_T)' K_T (A_T + C_T G_T)] y_{T-1} + c_{T-1},$$

ahol  $c_{T-1}$  egy az  $\varepsilon_T$  kovariancia-mátrixától függő konstans. Egy additív konstansról eltekintve tehát (1.11) bal oldala az

$$M(y_{T-1} - \tilde{y}_{T-1})' H_{T-1}(y_{T-1} - \tilde{y}_{T-1})$$

kifejezéssel egyenlő, ahol

$$H_{T-1} = K_{T-1} + (A_T + C_T G_T)' K_T (A_T + C_T G_T),$$

$$H_{T-1} \tilde{y}_{T-1} = K_{T-1} \hat{y}_{T-1} + (A_T + C_T G_T)' K_T (\hat{y}_T - b_T).$$

Ebből egyrészt az következik, hogy az  $y_T$ -re, illetve  $x_T$ -re vonatkozó megmondolás — az 1. lemma alkalmazása — most megismételhető  $y_{T-1}$ -re, illetve  $x_{T-1}$ -re (majd később  $y_{T-2}$ -re és  $x_{T-2}$ -re, s i. t.), másrészt pedig, az utolsó két egyenletből egy minden  $t$ -re általánosítható rekurzív összefüggés adódik, ha bevezetjük a következő jelöléseket:

$$H_T = K_T, \quad k_T = h_T = H_T \hat{y}_T$$

$$h_{T-1} = H_{T-1} \tilde{y}_{T-1}, \quad k_{T-1} = K_{T-1} \hat{y}_{T-1}.$$

Ezekkel a jelölésekkel (1.9) tetszőleges  $t$ -hez tartozó megfelelője a következő lesz:

$$(1.12) \quad G_t = -(C_t' H_t C_t)^{-1} C_t' H_t A_t$$

$$q_t = (C_t' H_t C_t)^{-1} C_t' (h_t - H_t b_t - H_t \varepsilon_t).$$

Ezzel tehát az (1.1) kifejezés lineáris modell melletti minimalizálásának algoritmusát meg van határozva: a lineáris visszacsatolásban szereplő mátrixokat és vektorokat a  $t = T, T-1, T-2, \dots$  sorrendnek megfelelően kell kiszámítani, majd ezek birtokában az endogén és a szabályozó változók értékét a  $t = 1$  időponttól kezdve, az időben előre haladva lehet meghatározni. Bár a módszerrel kapcsolatos számítástechnikai tapasztalatainkat a dolgozat 3. részében fogjuk ismertetni, már itt megjegyezzük, hogy ez az egyszerűsége révén elegáns eljárás (1.12) valamint a  $H_t$ -re, illetve  $h_t$ -re vonatkozó rekurzív formulák következtében „elrontja” a modellben szereplő  $A_t$  és  $C_t$  mátrixok „ritkás” tulajdonságát, és már néhány iteráció után sűrű, teli  $H_t$  és  $G_t$  mátrixokat generál. Úgy tűnik, hogy ezen a számítástechnikai szempontból kedvezőtlen adottságon nemigen lehet segíteni.

A következőkben G. C. Chow-nak a nemlineáris modellek szabályozására adott algoritmusát ismertetjük vázlatosan. Feladatunk tehát az

$$M \sum_{t=1}^T (y_t - \hat{y}_t)' K_t (y_t - \hat{y}_t)$$

kifejezés minimalizálása az

$$(1.2) \quad y_t = g(y_t, y_{t-1}, x_t, w_t) + \varepsilon_t, \\ t = 1, 2, \dots, T$$

modell mint feltételrendszer mellett. Az eljárás a következő.

*Inicializálás:* kiindulópontunk a szabályozó változók valamilyen  $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_T$  kezdeti értéke.

1. lépés. A szabályozó változók adott értéke mellett  $t = 1, 2, \dots, T$  esetén Gauss-Seidel módszerrel meghatározzuk az endogén változók olyan  $y_1^0, y_2^0, \dots, y_T^0$  értékét, amelyek kielégítik az (1.2) nemlineáris modellt  $\varepsilon_t = 0$  esetén.
2. lépés. Linearizáljuk a modellt az  $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_T, y_1^0, y_2^0, \dots, y_T^0$  megoldás körül, és a linearizált modellt átrendezéssel hozzuk a következő alakra:

$$y_t = A_t y_{t-1} + C_t x_t + b_t + \varepsilon_t, \quad t = 1, 2, \dots, T.$$

3. lépés.

A célfüggvényt minimalizáljuk a lineáris modell mellett. Legyen az optimális szabályozás  $x_1, x_2, \dots, x_T$ ; ha ez bizonyos hibahatáron belül megközelíti  $\tilde{x}_1$ -et,  $\tilde{x}_2$ -t,  $\dots$ ,  $\tilde{x}_T$ -t, akkor készen vagyunk. Ellenkező esetben legyen  $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_T$  új értéke rendre  $x_1, x_2, \dots, x_T$ , és folytassuk az eljárást az 1. lépésnél.

## 2. Determinisztikus diszkrét szabályozási feladatok megoldása az általánosított redukált gradiensek módszerével

A bevezetésben mondottaknak megfelelően ebben a fejezetben a következő programozási feladat megoldásával foglalkozunk:

$$g(y_t, y_{t-1}, \dots, y_{t-\tau}, x_t, x_{t-1}, \dots, x_{t-\theta}) = 0, \\ t = 1, 2, \dots, T;$$

$$y_t \leq y_t \leq \bar{y}_t, \quad x_t \leq x_t \leq \bar{x}_t,$$

$$f(y_1, y_2, \dots, y_T, x_1, x_2, \dots, x_T) = \min!$$

Az itt alkalmazott jelölések értelemszerűen azonosak azokkal, amelyeket a dolgozat bevezetésében használtunk. Mint arra az előzőekben utaltunk, A. DRUD a tekintett feladat megoldására az általánosított redukált gradiensek módszerét [1], [9], [24] stb. látta a leginkább célravezetőnek, és ennek alapján dolgozta ki a CONOPT programcsomagot, melynek elvi alapjait a következőkben ismertetjük. Érdeemes megjegyezni, hogy a CONOPT első változata a hetvenes évek végén készült el, amikor a matematikai programozás művelői között általános volt az a vélemény, hogy a nagyméretű nemlineáris programozási feladatok numerikus megoldásának a leghatékonyabb eszköze, kivált nagy méretek esetén, az általánosított redukált gradiensek módszere. Az utóbbi években azonban, többek között K. SCHITTROWSKI különböző programozási módszereket összehasonlító munkája [22] nyomán, előtérbe kerültek az ún. SLC eljárások (SLC = sequence of linearly constrained problems), amelyek az



általános nemlineáris programozási feladatok megoldását lineárisan korlátozott és többnyire kvadratikus célfüggvényű részfeladatok sorozatának megoldására vezetik vissza. Az általánosított redukált gradiensek módszerének fontosságával kapcsolatos elképzelések ezáltal módosultak, ami természetesen nem azt jelenti, hogy ezek a módszerek nem maradnak továbbra is rendkívül fontosak a matematikai programozási feladatok numerikus megoldása szempontjából. A CONOPT-tal kapcsolatban emellett azt is le kell szögezni, hogy folyamatosan tökéletesítik; ismertetésünk a legfrissebb verzió koncepcióját tartalmazza.

Az általánosított redukált gradiensek módszerének leírásához néhány új jelölést kell bevezetnünk, sőt, az eddigi jelöléseket is bizonyos mértékig módosítanunk kell. Mindenekelőtt a módszer szempontjából nincs különbség endogén és szabályozó változók között, és ezért az  $y_t$  és  $x_t$  vektorokat egyetlen vektorra vonjuk össze, s a továbbiakban ezt a vektort jelöljük  $x_t$ -vel. Így módon a fenti programozási feladat a következő alakot ölti:

$$(2.1) \quad g(x_t, x_{t-1}, \dots, x_{t-\tau}) = 0, \quad t = 1, 2, \dots, T,$$

$$(2.2) \quad \underline{x}_t \leq x_t \leq \bar{x}_t, \quad t = 1, 2, \dots, T$$

$$(2.3) \quad f(x_1, x_2, \dots, x_T) = \min !$$

$x_t$  most  $m$ -dimenziós, — ahol  $m$  a korábbi  $m + n$ -nek felel meg —  $g(\cdot)$  továbbra is  $n$ -dimenziós vektor. Emlékeztetünk arra, hogy — bár ezt a jelölés nem tükrözi —  $g$  nem csupán az  $x_t$  vektoron keresztül függhet  $t$ -től, és így pl. (2.1) magában foglalja a következő lehetőséget is:

$$A_{1t}x_t + A_{2t}x_{t-1} + \dots + A_{\tau+1,t}x_{t-\tau} = b_t, \\ t = 1, 2, \dots, T,$$

ahol  $A_{1t}, A_{2t}, \dots, A_{\tau+1,t}$  és  $b_t$  adott,  $t$ -től függő mennyiségek. (2.1), ill.  $g$  Jacobi-mátrixának jelölésénél természetesen nem feltételezhetünk kapcsolatot az argumentumok között, ezért  $x_t$  és késleltetett értékei helyére rendre  $u_{1-t}, u_{2-t}, \dots, u_{\tau+1-t}$ -et írunk; ezek értelemszerűen  $m$ -dimenziós vektorok, melyek között semmilyen kapcsolat sincsen. A  $(\tau + 1)m$ -dimenziós  $(x'_t, x'_{t-1}, \dots, x'_{t-\tau})'$  helyett röviden  $z_t$ -t írva mármost a következő jelölést vezetjük be:

$$G_j(z_t) = \left. \frac{\partial g}{\partial u_j} \right|_{z_t}$$

$$j = 1, 2, \dots, \tau + 1, \quad t = 1, 2, \dots, T.$$

A teljes (2.1) feltételrendszerhez tartozó Jacobi-mátrixot egyszerűen  $G$ -vel jelöljük, azaz, pl.  $\tau = 2$  és  $T = 4$  esetén  $G$  a következő képet mutatja:

$$G = \begin{bmatrix} G_1(z_1) & & & & & \\ G_2(z_1) & G_1(z_2) & & & & \\ G_3(z_1) & G_2(z_2) & G_1(z_3) & & & \\ & G_3(z_2) & G_2(z_3) & G_1(z_4) & & \end{bmatrix}.$$

Az általánosított redukált gradiensek módszerének, ill. a CONOPT-program elvi alapjainak leírásához az eddigieknél tömörebb írásmódra is szükségünk

lesz. A  $Tm$ -dimenziós  $(x'_1, x'_2, \dots, x'_T)'$  vektor helyére egyszerűen  $y$  betűt írva, (2.1)–(2.3) a következő alakú lesz:

$$(2.4) \quad F(y) = 0,$$

$$(2.5) \quad y \leq \bar{y} \leq \bar{y},$$

$$(2.6) \quad f(y) = \min!$$

Itt  $F(y)$  azt a hipervektort jelöli, melynek komponensei  $g(z_1), g(z_2), \dots, g(z_T)$ . A (2.1)–(2.3) rendszert a továbbiakban *dinamikus*, a (2.4)–(2.6) rendszert pedig *statikus modellnek* fogjuk nevezni. Az  $y$  vektor komponenseinek jelölésére az  $y_1, y_2, \dots, y_{mT}$  jeleket is fogjuk használni; ezek korábbi jelentése feltehetően nem fog zavart okozni. Eddigi jelöléseinket még a következőkkel kell kiegészítenünk:

$$h' = \frac{\partial f}{\partial y}, \quad h'_t = \frac{\partial f}{\partial x_t}, \quad t = 1, 2, \dots, T, \quad G(y) = \frac{\partial F(y)}{\partial y},$$

tehát a statikus rendszer Jacobi mátrixát ugyanúgy jelöljük, mint a dinamikus rendszerét. Az általánosított redukált gradiensek módszerének sémáját a következőkben a statikus modell jelöléseivel írjuk le.

1. Legyen  $y^0$  egy megengedett induló megoldás, és legyen  $k = 0$ .
2. Határozzuk meg az aktuális  $G = G^k$  Jacobi-mátrixot.
3. Válasszunk ki egy  $B$  bázist a következő feltételek mellett:
  - (a)  $G$ -nek a  $B$ -hez tartozó része,  $G_B$  legyen invertálható;
  - (b)  $y^k$ -nak  $B$ -hez tartozó részére,  $y_B^k$ -re teljesüljön az  $\underline{y}_B^k < y_B^k < \bar{y}_B^k$  egyenlőtlenség.

Legyen  $N$  a  $B$  komplementer halmaza:  $B \cup N = \{1, 2, \dots, mT\}$ . Az oszlopok, ill. komponensek sorrendjétől eltekintve fennállnak a  $G = (G_B, G_N)$  és a  $h' = (h'_B, h'_N)$  összefüggések.

4. Határozzuk meg a duál változók  $w' = h'_B G_B^{-1}$  vektorát és a  $d' = h'_N - h'_B G_B^{-1} G_N$  redukált gradienst.<sup>5</sup>
5. Ha minden olyan  $i \in N$  esetén, amelyre  $d_i \neq 0$ ,  $d_i > 0$  esetén  $\underline{y}_i^k = \underline{y}_i$ ,  $d_i < 0$  esetén pedig  $\underline{y}_i^k = \bar{y}_i$  teljesül, akkor  $y^k$  Kuhn-Tucker-pont és az eljárás befejeződött.
6. Legyen az  $S$  szuperbázis azoknak a változóknak a halmaza, amelyekre  $i \in N$  és  $d_i \neq 0$ , azonban  $d_i > 0$  esetén  $\underline{y}_i^k > \underline{y}_i$ , illetve  $d_i < 0$  esetén  $\underline{y}_i^k < \bar{y}_i$ . Az  $N \setminus S$  halmazt jelöljük továbbra is  $N$ -nel,  $y^k$  megfelelő részei pedig legyenek  $y_S^k$  és  $y_N^k$ . Keressünk  $d$  ismeretében olyan megengedett  $c = c^k$  irányt, melyre vonatkozóan  $f$  mint  $y_S$  függvénye  $y_S^k$  egy környezetében növekvő.
7.  $\alpha > 0$  különböző értékei mellett keressük a feladat egy új megengedett megoldását: rögzített  $\alpha$  esetén határozzuk meg a bázisváltozók  $y_B$  vektorát az  $F(y_B, y_S^k - \alpha c^k, y_N^k) = 0$  egyenletből Newton-módszerrel.
8. A 7. pontban meghatározott megengedett megoldások közül a legjobb cél-függvényértékű megoldást jelöljük  $y^{k+1}$ -gyel, növeljük  $k$  értékét 1-gyel, és folytassuk az eljárást a 2. lépésnél.

Mint a 2. Drud is megjegyzi, egy az általánosított redukált gradiensek módszerén alapuló gépi program hatékonysága nagymértékben függ attól, hogy a cél-függvény javításának irányát ( $c^k$ ) hogyan választjuk, milyen eljárást alkalma-

<sup>5</sup> A gradienst sorvektornak fogjuk fel, így  $d$ , a redukált gradiens transzponáltja, oszlopvektor.

zunk a választott irányban való keresésnél stb.; nagyméretű feladatok esetén emellett kitüntetett szerepe van annak, hogy milyen mértékben sikerül kihasználni a Jacobi mátrix ritkás tulajdonságát, ami általában az ilyen feladatok velejárója. A következőkben azokat a részalgoritmusokat, eljárásokat fogjuk áttekinteni, amelyek a CONOPT programcsomag hatékony működését biztosítják; ennek során azonban, főként terjedelmi szempontok miatt, egyes technikai megoldásokat részletesebben, másokat vázlatosabban ismertetünk. Többnyire a *statikus feladattal* fogunk foglalkozni, a *dinamikus feladatot* csupán olyan esetekben vizsgáljuk részletesen, amikor az eredeti — tehát a (2.1)–(2.3) — feladat strukturális sajátosságaiból adódó lehetőségek kiaknázása nem magától értetődő, ilyen eset például a bázis kiválasztása és a bázismátrix invertálása. A jobb áttekinthetőség érdekében a további mondanivalónkat alfejezetekre tagoljuk a következők szerint:

- bázis kiválasztása és invertálás statikus feladat esetén;
- bázis kiválasztása és invertálás dinamikus feladat esetén;
- megengedett irány meghatározása a redukált gradiens segítségével (statikus feladat);
- a célfüggvényérték javítása adott irány mentén (statikus feladat);
- induló megengedett megoldás meghatározása.

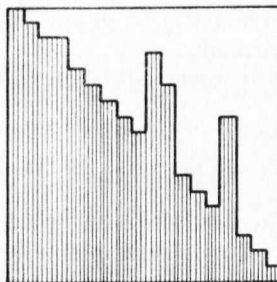
2.1. *Bázis kiválasztása és a bázismátrix invertálása statikus feladat esetén*

Ezt a részfeladatot a CONOPT programban ugyanúgy oldották meg, mint a kereskedelmi forgalomban levő lineáris programozási kódoknál, vagyis az inverz szorzatalakjának és a gyakorlatilag optimális pivotelem-kiválasztást biztosító HELLERMAN—RARICK-eljárás [15] felhasználásával. Mint ismeretes, ez a következőket jelenti:

- a) adott  $\bar{n} \times \bar{m}$ -es  $A$  mátrixból, feltéve, hogy  $\bar{n} < \bar{m}$  és  $\text{rang}(A) = \bar{n}$ , az eljárás kiválaszt  $\bar{n}$  független pivotelemet oly módon, hogy az ezek által meghatározott  $\bar{n} \times \bar{n}$ -es  $A_B$  bázismátrix a sorok és oszlopok sorrendjétől eltekintve a lehető legkisebb mértékben tér el egy alsó háromszögmátrixtól;
- b) az eljárás előállítja  $A_B$  inverzét, éspedig

$$(2.7) \quad A_B^{-1} = E_r E_{r-1} \dots E_2 E_1$$

alakban, ahol  $E_1, \dots, E_r$  ún.  $\eta$ -mátrixok, amelyek az egységmátrixtól csupán egy-egy oszlopban különböznek; ezeket az egységvektortól különböző oszlopokat  $\eta$ -vektoroknak nevezzük. Az alsó index itt a keletkezés sorrendjére utal,  $r = \bar{n}$  nem feltétlenül teljesül.



1. ábra

Az eljárás által kiválasztott  $A_B$  bázismátrix sorainak és oszlopainak alkalmas átrendezésével elérhető, hogy a pivotelemek a főátlóba kerüljenek, és akkor a mátrix az 1. ábrán látható képet mutatja; a vonalkázatlan részek csupa nullaelemnek felelnek meg, emellett természetesen lehetnek nulla-elemek a főátló alatt is. Azokat az oszlopokat, amelyekben a főátló felett csak nulla-elemek vannak, *trianguláris oszlopoknak*, a többit — tehát a „kilógó” oszlopokat — *extrém oszlopoknak* nevezzük; ez utóbbiak közhasználatú angol neve *spike*. A Hellerman — Rarick (a következőkben: HR) eljárás arra törekszik, hogy az extrém oszlopok minél rövidebbek legyenek, vagyis, hogy — az ábrán látható elrendezésben — minél kevesebb nullától különböző elemük legyen a főátló felett.

A HR-eljárás során a pivotelemek kiválasztása és  $A_B^{-1}$  szorzatalakban való előállítás, legalábbis a legtöbb gyakorlati megvalósítás esetén, egyidejűleg megy végbe. Ha történetesen  $A_B$ -ben a pivotelemek — úgy, mint az 1. ábrán — a főátló mentén helyezkednek el, akkor (2.7)-ben  $r = \bar{n}$ , bármely  $E_k$ -ban ( $1 \leq k \leq \bar{n}$ ) az  $\eta$ -vektor pontosan a  $k$ -adik helyen áll, és azt a következő képlettel lehet kiszámítani:

$$(2.8) \quad (\eta_1^k, \eta_2^k, \dots, \eta_n^k)' = \\ = \frac{1}{a_{kk}^{k-1}} (-a_{1k}^{k-1}, \dots, -a_{k-1,k}^{k-1}, 1, -a_{k+1,k}^{k-1}, \dots, -a_{nk}^{k-1})',$$

ahol

$$(a_{1k}^{k-1}, a_{2k}^{k-1}, \dots, a_{nk}^{k-1})'$$

az

$$E_{k-1}E_{k-2} \dots E_2E_1A_B$$

mátrix  $k$ -adik oszlopát jelöli ( $E_0A_B = A_B$ ). Ha ezt a mátrixot balról  $E_k$ -val megszorozzuk, akkor csak az *extrém oszlopok* transzformálódnak, a *trianguláris oszlopok* változatlanul maradnak. Ez a szabály az általános esetben is érvényes, és a fenti algoritmus, illetve a CONOPT-program szempontjából azt jelenti, hogy a Jacobi-mátrixból kiválasztott bázismátrixnak csupán az extrém oszlopaít kell transzformálni az  $\eta$ -mátrixok segítségével, a trianguláris oszlopokat nem; ez utóbbiak transzformáltja (2.8) alapján szükség esetén bármikor előállítható.

Az általános elvek alkalmazása mellett a CONOPT-ban az invertálás vonatkozásában még a további, többé-kevésbé speciális szempontok is érvényesültek. Először is, az invertálás kétféle „üzemmódját” alkalmazza a program, és pedig a *báziskeresést* és az *újrainvertálást*. Báziskeresésre — amely természetesen magában foglalja az invertálást is — akkor van szükség, ha

- induló bázist kell meghatározni,
- báziscserét kell végrehajtani, mert egy vagy több bázisváltozó elérte alsó v. felső korlátját,
- az újrainvertálás nem sikerült.

Újrainvertálásról akkor beszélünk, ha a bázisváltozók halmaza ugyanaz, mint az előző megengedett megoldás esetén volt, és csak a konstanstól különböző Jacobi-elemek értéke változott. Ebben az esetben az  $\eta$ -mátrixok, ill.  $\eta$ -vektorok struktúrája, azaz, nulla- és nem nulla elemeik száma és pozíciója is változatlan, csupán a nullától különböző elemek numerikus értéke változik. Figyelembe véve, hogy — mint előbb említettük — a HR eljárás szempontjából csak

a bázismátrix extrém oszlopaít kell transzformálni, az újrainvertálás a báziskereséshez viszonyítva igen gyorsá tehető. Célszerű tehát egy már meghatározott bázist addig használni, amíg csak lehetséges. A tapasztalatok szerint ez a lehetőség sok esetben adva van, ugyanis a (2.4)–(2.6) feladat nemlineáris voltából következően az irány menti keresések mintegy 80%-a belső pontban fejeződik be, amikor is báziscserére nincs szükség, s így újrainvertálást lehet alkalmazni.

A CONOPT-program készítése során célszerűnek bizonyult néhány módosítás alkalmazása a HR eljárás implementálásánál. A fontosabb változtatások a HR eljárás eredeti változatához képest a következők voltak:

- a bázis kiválasztása során a szóba jövő extrém oszlopoknál nem csupán a háromszögmátrixból való „kilógás” mértékét kell figyelembe venni, hanem a megfelelő változónak a korlátaitól mért távolságát is;
- a pivotelem kiválasztásánál numerikus toleranciát kell megadni az elem abszolút értékére vonatkozóan;
- a bázismátrix trianguláris oszlopaít elegendő megjelölni a Jacobi-mátrixban, az extrém oszlopokat azonban transzformálni kell, és az így adódó oszlopok logikai struktúráját ugyanúgy kell tárolni, mint a Jacobi-mátrix oszlopaít.

Ha báziskeresés vagy újrainvertálás során egy lehetséges pivotelem abszolút értéke nem éri el az adott toleranciaszintet, akkor a megfelelő oszlop nem kerülhet a bázisba, illetve újrainvertálás esetén ki kell hagyni a bázisból. Ekkor további oszlopokban kell pivotelemet keresnünk, ami azt is jelentheti, hogy bővítenünk kell a bázisba bevehető változók halmazát, vagyis enyhítjük a változóknak a hozzájuk tartozó korlátoktól mért eltéréseire vonatkozó megszorítást. Ha már semmiképpen sem lehet a bázisba bevonható változók körét bővíteni, akkor egy bizonyos határig csökkenteni lehet a pivotelem abszolút értékéhez tartozó alsó korlátot, ezen túl azonban az invertálást sikertelennek kell minősítenünk, azzal a következtetéssel, hogy a Jacobi-mátrix nem tesz eleget a rangra vonatkozó kikötésnek. A báziscserével kapcsolatban megjegyezzük, hogy az ugyanúgy megy végbe, mint lineáris programozás esetén: a belépő vektort transzformáljuk az  $\eta$ -mátrixokkal, majd — pozíciójának megjegyzésével —  $(r + 1)$ -edik elemként az  $\eta$ -vektorok közé soroljuk.

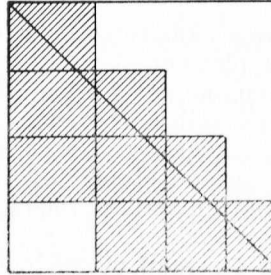
## 2.2. Báziskeresés és invertálás dinamikus feladat esetén

Természetesnek tűnik, hogy a (2.1)–(2.3) feladat speciális struktúráját célszerűen ki lehet használni báziskeresésnél és az inverznek szorzat alakban való előállításánál. Ezzel kapcsolatban mindenképp azt kell tisztáznunk, hogy mit állíthatunk általánosságban a bázismátrixok felépítéséről dinamikus feladatok esetén.

A feladat Jacobi-mátrixa, mint korábban láttuk, alsó blokkháromszögmátrix, melyben minden egyes blokk  $n$  sorból és  $m$  oszlopból áll. Kézenfekvőnek látszik a feltételezés, hogy minden egyes  $G_B$  bázismátrix is ilyen felépítésű, azzal az eltéréssel, hogy most a blokkok  $n \times n$ -es méretűek. Ha ez igaz lenne, akkor nagyon egyszerű, elegáns formában lehetne előállítani a bázis inverzeket, sajnos azonban a bázismátrixok felépítése olyan, mint ahogyan a 2. ábrán látható.

A bázisokat tehát az a körülmény jellemzi, hogy a főátló mentén elhelyezkedő blokkok jobb alsó sarka vagy a főátlón, vagy pedig a főátló fölött helyezkedik el.

A pivotelemek elhelyezkedése szempontjából ez a következőt jelenti: a korábban bevezetett jelöléseket használva nevezzük a teljes Jacobi-mátrix  $G_1(z_1)$ ,  $G_1(z_2), \dots, G_1(z_T)$  blokkjait *egyidejű* vagy *késleltetés nélküli* blokkoknak, az összes  $G_i(z_i)$  alakú blokkot pedig, ahol  $i \geq 2$ , *késleltetett* blokkoknak; ekkor azt mondhatjuk, hogy a pivotelemek egy része a késleltetett blokkokhoz tartozik.



2. ábra

Az eddigiek figyelembevételével a dinamikus báziskiválasztás és invertálás algoritmus  $t = 1, 2, \dots, T$  külső ciklusból áll, és a  $t$ -edik ciklus eredménye egy  $nt \times nt$  méretű bázismátrix kiválasztása és az inverznek szorzat alakban való előállítás. Az algoritmus 1. ciklusában a  $G$  Jacobi-mátrix első  $n$  sorából és első  $m$  oszlopából, más szóval: a  $G_1(z_1)$  blokkból a 2.1. alfejezetben mondottak figyelembe vételével kiválasztunk egy  $n \times n$ -es  $B_1$  bázismátrixot, és természetesen előállítjuk ennek inverzét is.

Az első  $t$  ciklus végrehajtása után az oszlopok sorrendjétől eltekintve  $G$  első  $n(t+1)$  sora és első  $m(t+1)$  oszlopa a 3. ábrán látható képet mutatja. A blokkok jelentése a következő:

$A_t$	$B_t$	0
$C_t$	$D_t$	$G_{1,t+1} = G_1(z_{t+1})$

3. ábra

$B_t$  az első  $t$  időszakhoz tartozó, már meghatározott bázismátrix inverze abban az értelemben, ahogyan az előző alfejezetben leírtuk (csak az extrém oszlopok transzformálására van szükség);  $B_t$  már tartalmazhat olyan pivotelemeket, melyek késleltetett blokkhoz tartoznak;

$D_t$  olyan részmátrix, melyet  $G$   $(nt+1)$ -edik,  $(nt+2)$ -edik,  $\dots$ ,  $n(t+1)$ -edik sorai és  $B_t$  oszlopai határoznak meg;

$A_t$  és  $C_t$  a  $G$  első  $n(t+1)$  sorának olyan oszlopaiból áll, melyek az első  $t$  időszakhoz tartoznak, és nem kerültek a bázisba;

$G_{1,t+1} = G_1(z_{t+1})$   $G$ -nek a  $(t+1)$ -edik időszakhoz tartozó, késleltetés nélküli blokkja.

A  $(t+1)$ -edik cikluson belül az 1. lépésben a  $G_{1,t+1}$  blokkon belül keresünk pivotelemeket a HR eljárással. Tegyük fel, hogy  $n^* < n$  elfogadható pivotelemet sikerül meghatározni; a hiányzó  $n - n^*$  számú pivotelemet *ideiglene-*



sen egyesekkel, a megfelelő oszlopokat pedig az egységmátrix oszlopaival pótoljuk. A 3. ábrán látható részmatrix ezután a 4. ábrán látható részmatrixba megy át; a vonalkázott  $(E_t, I)$  mátrix itt a kiválasztott  $n \times n$ -es mátrix inverzét reprezentálja,  $I$  most  $(n - n^*) \times (n - n^*)$  rendű egységmátrix  $\bar{G}_{1,t+1}$  a  $G_{1,t+1}$  mátrixnak az a része, amelynek oszlopai nem kerültek be a bázisba. Jelöljük azt a mátrixot, amely az ábrán vastagon bekeretezett részből  $D_t$ -nek

$A_t$	$B_t$	$O$	$O$
$C_t$	$D_t$	$E_t$	$\bar{G}_{1,t+1}$

4. ábra

—  $(E_t, I)D_tB_t$ -vel való helyettesítésével adódik,  $B_{t+1}$ -gyel. A továbbiakban báziscserét fogunk végrehajtani oly módon, hogy az egységmátrixhoz tartozó oszlopok helyére az  $A_t$  és  $C_t$  részmatrixokhoz tartozó oszlopok kerüljenek; ekközben mind a közbülső eredményeket, mind a végeredményt továbbra is  $B_{t+1}$ -gyel fogjuk jelölni

A  $(t + 1)$ -edik ciklus 2. lépése tehát a következő: legyen  $e_j$  az aktuális  $B_{t+1}$ -nek az az — egységmátrixhoz tartozó — oszlopa, amelyet  $(A'_t, C'_t)'$  valamelyik oszlopával szeretnénk kicserélni. Ebben az esetben, mint könnyen belátható, az  $e'_j B_{t+1} (A'_t, C'_t)'$  sorvektorban kell pivotelemet keresni. Ha megfelelő pivotelemet nem találunk, akkor a 2.1. alfejezetben megadott módon kell eljárni, egyébként következzen a 3. lépés.

A  $(t + 1)$ -edik ciklus 3. lépésében a pivot oszlopot transzformálni kell  $B_{t+1}$ -gyel, ezáltal megkapjuk  $B_{t+1}$  új értékét, és amennyiben a mátrix több egységmátrixhoz tartozó oszlopot már nem tartalmaz, a ciklust befejeztük; ellenkező esetben az algoritmus a 2. lépésnél folytatódik.

Az eljárásból következően a  $T$ -edik külső ciklus után  $B_T = G_B^{-1}$ .

### 2.3. Megengedett irány meghatározása a redukált gradiens ismeretében

A célfüggvény javításának irányával kapcsolatban a (2.4)–(2.5) statikus feladat vizsgálatára szorítkozunk, mivel az egyetlen dolog ezzel kapcsolatban, amiben a feladat, illetve a bázismátrix dinamikus struktúrája szerepet játszik, a duál változók, ill. a redukált gradiens kiszámítása  $w' = h'_B G_B^{-1}$ , ill.  $d' = h'_N - w' G_N$  alakban. Erre vonatkozóan elegendő azt megjegyeznünk, hogy a (2.7) képlet értelmében a  $h'_B$  vektort az  $\eta$ -mátrixokkal jobbról a következő sorrendben kell megszorozni:  $E_r, E_{r-1}, \dots, E_2$  és  $E_1$ . A CONOPT egyébként az inverz numerikus stabilitásának ellenőrzése céljából a  $h'_B - w' G_B$  vektort is kiszámítja, melynek elvben nullával kell egyenlőnek lennie.

Az általánosított redukált gradiens módszerének fenti algoritmusában a bázisba nem tartozó változókat *szuperbázisra* ( $S$ ) és ezen kívüli változók halmazára ( $N$ ) bontottuk fel (6. lépés). Az aktuális megengedett megoldás mellett az  $f$  célfüggvényt a szuperbázis-változók függvényének tekintjük, és így

amennyiben sikerül több, egymást követő megengedett megoldás esetén ugyanazt a bázis-szuperbázis-halmazt használnunk, akkor ezekben a lépésekben gyakorlatilag feltétel nélküli szélsőérték-feladattal van dolgunk. Az első megengedett megoldásnál az  $S$  szuperbázist úgy definiáljuk, mint azoknak a bázisba nem tartozó  $y_i$  változóknak a halmazát, amelyekre  $\underline{y}_i < y_i < \bar{y}_i$ , és a továbbiakban a következők szerint módosítjuk:

- $y_i$  kilép a szuperbázisból, ha elérte alsó vagy felső korlátját, illetve, ha belép a bázisba, valamely  $y_j$  változó helyére, amely elérte alsó vagy felső korlátját;
- $y_i$  belép a szuperbázisba, ha azokra a  $j \in N$  indexekre, amelyekre  $y_j^k - \alpha d_j$  kellőképpen kis pozitív  $\alpha$ -értékek mellett  $\underline{y}_j$  és  $\bar{y}_j$  közé esik,  $d_i$  — a redukált gradiens megfelelő komponense — abszolút értékben maximális, és ezen túlmenően még egy dinamikus változó küszöbértéket is meghalad.

A CONOPT célfüggvény javítását kvázi-Newton módszerrel végzi, az egymást követő keresési irányok meghatározására pedig a C. G. BROYDENTÓL [3], R. FLETCHERTŐL [11], D. GOLDFARBÓL [13] és D. F. SHANNOTÓL [23] származó BFGS formulát alkalmazza. Abban az esetben, amikor az algoritmus működésének egy szakaszában változatlan bázis-szuperbázis halmazzal dolgozunk, ez a következőket jelenti. Legyen

$\Phi = \Phi(y_S) = f(y_B(y_S), y_S, y_N)$  a célfüggvény mint a szuperbázis változóinak a függvénye;

$y_S^0, y_S^1, y_S^2, \dots$  a szuperbázis változók vektorának egymás utáni értékei változatlan  $S$  szuperbázis mellett;

$d^0, d^1, d^2, \dots$  a redukált gradiens, vagyis a  $\Phi$  függvény  $y_S$  szerinti gradiensek egymást követő értékei;

$c^0, c^1, c^2, \dots$  a célfüggvény javítása irányának egymás utáni értékei;

$M_0, M_1, M_2, \dots$  a  $\Phi(y_S)$  függvény Hesse-féle mátrixának ( $M = \partial^2 \Phi / \partial y_S^2$ ) egymást követő approximációi;  $M_0$  vagy az egységmátrixszal egyenlő, vagy pedig egy korábbi szuperbázis halmazhoz tartozó  $M$  mátrixból alkalmas transzformációval áll elő (1. a későbbiekben).

Ezekkel a jelölésekkel az eljárást a következőképpen írhatjuk le.  $k = 0, 1, 2, \dots$  esetén legyen

$$c^k = M_k^{-1} d^k;$$

$$y_S^{k+1} = y_S^k - \alpha_k c^k, \text{ ahol } \alpha_k \text{ a } \Phi(y_S^k - \alpha c^k) \text{ függvény}$$

egy minimumhelye;

$$(2.9) \quad v^k = y_S^{k+1} - y_S^k, \quad u^k = d^{k+1} - d^k;$$

$$(2.10) \quad M_{k+1} = M_k + \frac{u^k (u^k)'}{(u^k)' v^k} - \frac{M_k v^k (v^k)'}{(v^k)' M_k v^k}. \quad (\text{BFGS formula})$$

Abban az esetben, amikor  $\Phi$  szigorúan konvex kvadratikus függvény, a kvázi-Newton módszernek ez a változata  $n^*$  lépésen belül megtalálja az optimumhelyet, feltéve, hogy az  $\alpha_k$  értékek meghatározása pontos eljárással megy végbe [11];  $n^*$  itt az  $y_S$  komponenseinek a számát jelöli.

A Hesse-mátrix  $M_k$  közelítése hasznos információt tartalmaz a célfüggvény viselkedéséről, és ezért célszerű ebből az információból minél többet megőriznünk abban az esetben, amikor a bázis- és szuperbázis változók halmazának

megváltoztatására van szükség. A CONOPT-ban a következő megoldást alkalmazták:

- (i) ha egy új változó belép a szuperbázisba, akkor  $M_k$  egy sorral és egy oszloppal bővül: az új diagonálemet az eddigi diagonálemek mértani közepével egyenlőnek választjuk, az új sor és oszlop többi eleme kezdetben nulla;
- (ii) ha egy változó kilép a szuperbázisból, akkor a megfelelő sort és oszlopot kihagyjuk  $M_k$ -ből;
- (iii) ha a  $j$ -edik szuperbázis változó helyét cserél az  $i$ -edik bázis változóval, akkor két esetet különböztetünk meg: ha több ilyen  $i, j$  indexpár van, akkor  $M_0$  értékét az új szuperbázisban az egységmátrixszal tesszük egyenlővé, míg az egy ilyen indexpár esetére vonatkozó teendőket a következőkben részletezzük.

Eljárásunknak az az alapja, hogy feladatunk  $F(y) = 0$  feltételrendszerét az aktuális  $y^k$  megengedett megoldás körül linearizáljuk. Ekkor, mint könnyen belátható, a régi szuperbázis változók ( $y_S$ ) és az új szuperbázis változók ( $\bar{y}_S$ ) között a következő kapcsolat áll fenn:

$$E\bar{y}_S = y_S,$$

ahol  $E$  most olyan transzponált  $\eta$ -mátrix, amelynek  $j$ -edik sora

$$(\eta_1, \eta_2, \dots, \eta_{n^*}) = -\frac{1}{\gamma} e_i' G_B^{-1} G_S + \frac{\gamma - 1}{\gamma} e_j'$$

és

$$\gamma = e_i' G_B^{-1} G_S e_j;$$

$e_i$  és  $e_j$  az  $n^* \times n^*$  méretű egységmátrix  $i$ -edik, ill.  $j$ -edik oszlopa. A Hesse-mátrix közelítésének transzformálását mármost a következőképpen kell elvégezni:

- (a) ki kell számítani a redukált gradienst az új szuperbázis-térben, legyen ez  $\bar{d}'$ ;
- (b) a redukált gradienst vissza kell transzformálni a régi szuperbázis-térbe:

$$(d^{k+1})' = d' = \bar{d}' E^{-1},$$

majd a (2.9)–(2.10) képletekkel kiszámítjuk a Hesse-mátrix  $M_{k+1}$  közelítését;

- (c) a mátrixot transzformáljuk az új szuperbázis-térbe:

$$\bar{M}_{k+1} = E' M_{k+1} E,$$

majd elhagyjuk azt a változót, amely a báziscserét szükségessé tette.

#### 2.4. A célfüggvényjáték javítása adott egyenes mentén

Ezzel, az általánosított redukált gradiensek módszere szempontjából kulcsfontosságú, részfeladattal kapcsolatban a következő kérdéseket fogjuk megvizsgálni:

- a Newton-módszer implementálásának szempontjai, különös tekintettel a hatékonyságra;
- a lépésköz korlátjának becslése a bázis- és szuperbázis változók egyéni korlátainak függvényében;

- az adott egyenes mentén való keresés befejezésének kritériumai;
- a numerikus pontatlanságból származó eredménytelenség kivédésének módszerei.

A Newton-módszer alkalmazására a bázis változók meghatározásához, tehát az

$$F(y_B^k, y_S^k - \alpha c^k, y_N^k) = 0$$

egyenletnek  $y_B^k$ -re való megoldásához van szükségünk, a bázison kívüli változók  $y_N^k$ , a szuperbázis változók  $y_S^k$ , a keresési irány  $c^k$  és a lépésköz  $\alpha$  rögzített értéke mellett. A következőkben  $y_S^k - \alpha c^k$ ,  $y_N^k$  helyett egyszerűen  $y_N$ -et,  $y_B^k$  helyett  $y_B$ -t írunk. A Newton-módszer elvben a következőket jelenti:

- kiindulunk egy  $y_B = y_B^0$  induló megoldásból (ennek megválasztására még visszatérünk);
- $l = 0, 1, 2, \dots$  esetén az

$$(2.11) \quad y_B^{l+1} = y_B^l - G_B^{-1} F(y_B^l, y_N)$$

iterációt alkalmazzuk egy alkalmasan választott megállási kritérium teljesüléséig.

Ezen a ponton röviden utalunk a *dinamikus feladattal* kapcsolatos legfontosabb tudnivalókra.  $G_B^{-1}$  minőségében ugyanazt a mátrixot használjuk, amely a redukált gradiens kiszámításánál szerepelt (l. az erre vonatkozó megjegyzést a 2.3. alfejezet elején); az ott követett eljárással ellentétben most az  $F$  vektort az  $\eta$ -mátrixokkal a következő sorrendben kell megszorozni:  $E_1, E_2, \dots, E_{r-1}$  és  $E_r$ . Ehhez még azt kell megjegyeznünk, hogy annak megfelelően, ahogyan a dinamikus bázis inverz,  $G_B^{-1} = B_T$  létrejött (2.2. alfejezet), most a (2.11) iterációs képlet alkalmazása  $t = 1, 2, \dots, T$  külső ciklus, és ezeken belül  $l$  szerinti *belső* ciklusok elvégzését fogja jelenteni. Mivel a bázismátrixok nem igazi blokk-háromszögmátrixok, hanem olyanok, mint a 2. ábrán látható mátrix, ez azt jelenti, hogy  $t_1 < t_2$  esetén a  $t_2$ -edik külső ciklus elronthatja az előző eredményét,  $G_B$  azon oszlopai ugyanis, amelyek valamely korábbi időszakhoz tartoznak, pivotelemük viszont a  $t_2$ -edik időszaknak megfelelő sorok valamelyikében van, módosítják az  $y_B$ -nek  $t = 1$  és  $t = t_2 - 1$  közötti részére kapott eddigi értékét. Amennyiben ezáltal egyes változók jelentős mértékben túllépnek alsó vagy felső korlátjukon, a Newton-módszer alkalmazását előlről kell kezdeni, kisebb lépéshossz alkalmazása mellett.

A továbbiakban mármost csak a (2.4)–(2.6) statikus feladattal foglalkozunk. A CONOPT-ba a divergencia, illetve az abból adódó fölösleges számítások kivédésére a következő kritikus értékek vizsgálatát építették be:

$l$  az iterációk száma;

$\delta_1 = F(y_B, y_N)$  normája, a komponensek abszolút értékének összege;

$\delta_2 = \sum_i w_i g_i = h'_B G_B^{-1} F$ , a célfüggvényértéknek a korlátozó feltételek megsértéséből származó hibája (becslés);

$\delta_3 = \max_i \{(y_i - \underline{y}_i)/(1 + |\underline{y}_i|), (\bar{y}_i - y_i)/(1 + |\bar{y}_i|)\}$ , a korlátok megsértésének mértéke.

Ezekhez a mennyiségekhez rendre a következő felső korlátok tartoznak:  $l_{\max}$ ,  $\varepsilon_1$ ,  $\varepsilon_2$ , illetve  $\varepsilon_3$ . Ahhoz, hogy a Newton-módszer működését sikeresnek könyveljük el, a következő feltételeknek kell teljesülniök:

$$l \leq l_{\max}, \delta_1 \leq \varepsilon_1, \delta_2 \leq \varepsilon_2, \delta_3 \leq \varepsilon_3,$$

továbbá, hogy

$$(\delta_1^{l+1}/\delta_1^l)^{(l_{\max}-l+1)/2} < \varepsilon_1, \quad (\delta_2^{l+1}/\delta_2^l)^{(l_{\max}-l+1)/2} < \varepsilon_2,$$

ahol az  $l$  felső index az iteráció sorszámára utal; ez a két utóbbi kikötés a túlságosan lassú konvergencia ellen nyújt védelmet.  $\varepsilon_1$  és  $\varepsilon_2$  alapértéke  $n \cdot 10^{-3}$ , ill.  $10^{-5}$ . Ezeket kívülről meg lehet változtatni; kísérletek folynak arra, hogy maga a program változtassa dinamikusan ezeket a küszöbértékeket, azonban végérvényes megoldás ebben a részletkérdésben még nincsen.  $\varepsilon_2$  dinamikusan változik, és pedig a célfüggvény adott irányban való várható változása becslésének századrészétől és a célfüggvény egy abszolút alsó korlátjától függ. A  $\delta_2 \leq \varepsilon_2$  követelmény ennél fogva olcsó és pontatlan Newton-megoldások elfogadását teszi lehetővé, ha még távol vagyunk az optimumtól, és pontos megoldások kiszámítását eredményezi az optimum közelében.

A fölsőleges számolás elkerülésének egyik fontos lehetősége a lépéshossz korlátozása annak érdekében, hogy a Newton-módszerrel meghatározott új megoldások a megengedett tartományon belül maradjanak. Az  $y^k$  aktuális megoldásnál  $y_s^k$  a szuperbázis változók vektora és  $c^k$  a célfüggvény javításának iránya. A lépéshossz egy  $\alpha^*$  felső korlátja adódik az

$$y_s \leq y_s^k - \alpha c^k \leq \bar{y}_s$$

egyenlőtlenségből, az  $\alpha \leq \alpha^*$  lépéshossz azonban még nem garantálja a bázis változók megengedettségét. A CONOPT ezért extrapolálja a bázisváltozók értékét, és ennek figyelembe vételével határozza meg a lépéshossz  $\alpha_{\max}$  végleges korlátját. Az extrapoláció az irány menti keresés kezdetekor lineáris, az  $y_B^k$  kezdőpont és a  $c_B^k = -G_B^{-1}G_S c^k$  irány alapján, a későbbiekben pedig kvadratikusan, a  $c_B^k$  irány és két pont, vagy az időrendben három utolsó pont alapján.

A kezdeti lépéshosszt a CONOPT-ban a célfüggvényérték várható megváltozásának és a  $c^k$  keresési irány meredekségének hányadosával határozzák meg, ha ez az előbb meghatározott felső korláton belül van; a célfüggvényérték várható változása korábbi változások mértani közepe, a meredekség pedig a redukált gradiens és  $c^k$  skalár szorzatával egyenlő. A tekintett irányban való keresés a Newton-módszer ismételt alkalmazásából áll, és ennek megfelelően a következő esetek következhetnek be:

- a Newton-módszer alkalmazása sikeres volt, új megengedett megoldást határoztunk meg, amelyre  $f(y) < f(y^k)$ ;
- a Newton-módszer alkalmazása divergencia vagy túl lassú konvergencia miatt sikertelen volt;
- a Newton-módszer alkalmazása sikertelen volt, mert valamilyen változó lényegesen túllépte alsó vagy felső korlátját.

Az (a) esetben tovább folytatjuk az adott egyenes mentén a keresést, és az új lépéshosszt kvadratikusan interpoláció alapján határozzuk meg; ez azonban nem lehet nagyobb, mint az előző lépéshossz  $\hat{\alpha}$ -szorosa, ahol  $\hat{\alpha}$  kezdeti értéke 4. A keresésnek határt szab, ha

- egy lépésben a célfüggvényérték javulása kisebb, mint az eddigi javulás 20%-a;
- bekövetkezik a (b) és (c) esetek valamelyike.

A (b) esetben követendő eljárás attól függ, hogy az aktuális irány menti keresés során találtunk-e már  $y^k$ -nál jobb célfüggvényértékű megengedett megoldást, vagy nem. Ha igen, akkor csökkentjük a lépéshosszt, és  $\hat{\alpha}$  értékét a

négyzetgyökével helyettesítjük, majd újból próbálkozunk a Newton-módszer alkalmazásával. Ha ez a módosítás kétszer egymás után alkalmazva sem hoz eredményt, az adott irány menti keresést befejezzük. A másik esetben, vagyis ha még nem találtunk  $y^k$ -nál jobb, új megengedett megoldást, akkor a lépéshosszt  $\alpha$ -pal elosztjuk, és így próbálunk új megengedett megoldást meghatározni. Ily módon előbb-utóbb bekövetkezik az (a) eset, vagy pedig a lépéshossz egy kritikus érték alá csökken, amikor is speciális kivezető utat kell találnunk; rövidesen ezzel a kérdéssel is foglalkozunk.

A (c) esetben egységesen  $9/10$ -részt csökkentjük a lépéshosszt és így kísérjük meg a Newton-módszer ismételt alkalmazását. Ez biztosítja azoknak az eseteknek a numerikus szempontból való helyes kezelését, amelyekben az egyenes mentén való keresésnek a megengedett tartomány határa szab korlátot.

Az egyenes mentén való keresés során előfordulhat, hogy a Newton-módszer konvergenciájának biztosítása érdekében olyan rövid lépéshosszal kell dolgoznunk, hogy a célfüggvény javulásának mértéke nagyságrendben azonos a numerikus számítás pontatlanságából adódó torzítással. Ilyenkor a CONOPT a következő speciális szabályok alkalmazásával próbálja kimozdítani a számítást a holtpontról.

A: visszatérünk az előző iterációhoz és a pontossággal kapcsolatos toleranciát tizedrészére csökkentve, újra meghatározzuk az  $y^k$  megoldást;

B: a javított  $y^k$  pontban újra kiszámítjuk a  $G_B^{-1}$  inverzet, és új keresési irányt határozzunk meg;

C: a minimum-keresésben áttérünk egyszerű gradiens módszerre;

D: bővítjük a szuperbázist, ha lehetséges;

E: a pivotelemekre vonatkozó tolerancia növelésével új bázisra térünk át.

Ezeket a szabályokat egymás után kell alkalmazni, tehát a B szabályt csak akkor, ha A nem vezetett célhoz, a C szabályt csak olyankor, ha B sem segített. Ha még E után sem oldódott meg a probléma, akkor még egyszer megismételjük az egészet A-tól E-ig, és újabb sikertelenség esetén a program ilyen értelmű üzenettel abbahagyja a számítást.

### 2.5. Induló megengedett megoldás meghatározása

Ezt a feladatot a legtöbb nemlineáris programozási algoritmus a lineáris programozásban szokásos módon oldja meg, az ún. 1. fázis keretei között. Ennek, mint ismeretes, az a lényege, hogy egy az eredeti feladat szempontjából nem megengedett  $x$  megoldáshoz egy olyan módosított feladatot rendelnek, melynek  $x$  megengedett megoldása, és amelynek optimális megoldása szükség-szerűen megengedett az eredeti feladatra nézve. Ezt a módosított feladatot oldják meg a rendelkezésre álló eljárással az 1. fázisban.

A CONOPT-ban induló megengedett megoldás meghatározására más módszert alkalmaztak, és pedig a Newton-módszer egy változatát. Szokásos jelöléseink segítségével ezt az eljárást a következőképpen lehet leírni.

1. lépés. Válasszunk egy  $y$  induló megoldást (a gyakorlati feladatok többségénél kézenfekvő ennek megadása).
2. lépés. Válasszunk ki egy  $B$  bázist a  $HR$  eljárással, emellett természetesen  $y_B$ -re teljesüljön az  $y_B < y_B < \bar{y}_B$  egyenlőtlenség.
3. lépés. Számítsuk ki a bázisváltozók növekményét a Newton-módszernek megfelelően:  $\Delta y_B = -G_B^{-1}F(y_B, y_N)$ .



4. lépés. Ha  $y_B + \Delta y_B$  a bázis változókhoz tartozó korlátok közé esik, legyen ez a kifejezés  $y_B$  új értéke, és következzen a 3. lépés.
5. lépés. Legyen  $\alpha$  az a legnagyobb valós szám, amelyre  $y_B + \alpha \Delta y_B$  még nem sérti meg az alsó és felső korlátokat, és legyen  $y_B$  új értéke ez a kifejezés. Cseréljük ki azt a bázis változót, amely most alsó v. felső határon van, egy olyan változóra, amely az eljárás folytatásakor nem ütközik egyik korlátjába sem. Több lehetőség esetén olyan változót vonjunk be a bázisba, amely a pivotelem nagysága és a korlátoktól való távolsága alapján a legkedvezőbb. Számítsuk ki az új  $G_B^{-1}$  mátrixot, és következzen a 3. lépés.

Az algoritmus konvergenciáját a CONOPT természetesen hasonló módon ellenőrzi, mint ahogyan azt a 2.4. alfejezetben leírtuk. Az eljárás megengedett megoldások hiányának jelzésével fejeződik be akkor, ha az 5. lépésben nem sikerül olyan báziscserét végrehajtani, amely a folytatást lehetővé tenné. Az induló bázis megoldás meghatározásának ez a módja A. Drud tapasztalatai szerint számos esetben nagyon jól bevált, viszont bizonyíthatóan eredménytelenül működik olyan esetekben, amikor az  $F(y)$  függvény erősen nemlineáris a megengedett tartomány határának közelében. Kutatás folyik abban a vonatkozásban, hogy hogyan lehetne a CONOPT-ban ezt az eljárást még hatékonyabb, szélesebb körben alkalmazható eljárássá helyettesíteni.

### 3. A szabályozási programcsomagok hazai alkalmazásának tapasztalatai

Mind Chow szabályozási eljárásáról, mind pedig az általánosított redukált gradiensek módszerén alapuló CONOPT programról elmondhatjuk, hogy egyrészt gondosan tesztelték őket, másrészt több esetben alkalmazták konkrét gazdasági feladatok elemzésére. A Chow-féle program — melynek neve OPTNL — alkalmazásaira az I. fejezetben utaltunk; a CONOPT-tal kapcsolatban most csak azt emeljük ki, hogy számos tesztfeladat mellett olyan gyakorlati feladatokat is sikeresen megoldottak vele, melyekben az ismeretlenek és a korlátozó feltételek száma ezren felül volt. „A világ kávé-modellje”-nek [4] egyik változatában például 80 időszak, időszakonként 35 változó és 34 korlátozó feltétel szerepelt, s így az összes változó és feltétel száma 2800, ill. 2720 volt, és a program a CDC Cyber 176 számítógépen 334,7 CPU másodperc alatt talált meg egy optimális megoldást. Eredményesen alkalmazták a CONOPT-ot a kanadai gazdaság 35 egyenlethől álló CLEAR nevű negyedéves ökonometria modelljének szabályozására is [21]. A CONOPT-tal kapcsolatos további külföldi számítástechnikai tapasztalatok ismertetése megtalálható [8]-ban; a következőkben a hazai tapasztalatokról számolunk be. Ezek jelenleg még egyértelműen a kezdeti stádium jegyeit mutatják, megítélésünk szerint azonban így is alkalmasak arra, hogy a potenciális jövőbeni felhasználók figyelmét felhívják a lehetőségekre.

A CONOPT és az OPTNL alkalmazásával két ökonometria modellt szabályoztunk az MTA Közgazdaságtudományi Intézetének Ökonometria Laboratóriumában. Ugyanitt készültek a szabályozási számítások kiinduló pontjául szolgáló ökonometria modellek is, ezek főbb jellemzői a következők.

A kisebb méretű modell a magyar gazdaság legfontosabb összefüggéseit teljesen aggregált formában, egyetlen szektor feltételezésével írta le. A szimultán összefüggések rendszerét alkotó egyenletekkel nyomon követhető a pénzbefek-

tetéseknek a beruházások üzembe helyezésére, azon keresztül pedig a termelésre gyakorolt hatása. A bérek és más lakossági jövedelmek határozzák meg a fogyasztást, ez utóbbi, a termeléssel együtt magyarázza a külkereskedelem alakulását. A gazdálkodás eredményességét mutatja a nyereség nagysága, s az egyszersmind visszahat a vállalati beruházási lehetőségekre. A modell időszakonként

31 egyenletet,  
31 endogén  
22 egzogén és  
8 szabályozó változót

tartalmazott. A fontosabb szerepet játszó szabályozó változók között szerepelt a közösségi fogyasztás, a lakossági vásárolt fogyasztás deflátorá, a bruttó nemzeti termék deflátorá és az állami döntéskörbe tartozó beruházások. A modellel végzett kísérleti jellegű számításoknak a célja az volt, hogy olyan növekedési pályát generáljon a szabályozó változókon keresztül, amely biztosítja a nem rubel elszámolású fizetési mérleg megfelelő alakulását, nem sérti a modellben megfogalmazott gazdasági összefüggéseket, és mindenekelőtt — a lehetőségek határain belül — egyenletes növekedést tesz lehetővé. Ennek a modellnek a leírása és a szabályozási számítások eredményei — amelyek az OPTNL felhasználásával készültek, 4 időszakra — [16]-ban találhatóak.

Az első kísérlet tapasztalatai az ökonometriai modellek szabályozásának további alkalmazására ösztönöztek. A rendelkezésre álló számítógépi programok a jelek szerint a korábbinál valamelyest nagyobb feladat megoldására is alkalmasnak tűntek. Ennek alapján elkészült egy olyan — az előbb ismertetetthez sok tekintetben hasonló — modell, amely a termelésben öt, a külkereskedelemben három ágazatra bontotta a gazdaság tevékenységét, és így részletesebb vizsgálatra nyílt lehetőség. A modell kialakításánál célszerűen ki lehetett használni azt, hogy nem előrejelzés, hanem szabályozás céljára készült, és így lehetséges volt néhány, a modellezés során gyakran felvetődő problémának — mint pl. a termelés kinálati és keresleti oldalról való meghatározása közötti összhang megteremtésének — a szokásosnál egyszerűbb megoldása a szabályozás időhorizontján. A szabályozási feladat időszakonként

45 egyenletet,  
45 endogén  
27 egzogén és  
13 szabályozó változót

tartalmazott. A modell méreteivel együtt erősödött annak nemlineáris jellege is: az egyenletek közül kilenc volt nemlineáris, ezekben egyébként egyszerű, alacsony fokszámú polinomok szerepeltek. A szabályozó változók köre az ágazati beruházásokkal és az aktív keresők ágazatonkénti számával bővült, a bruttó termelés deflátorát pedig felváltotta a hozzáadott érték deflátorá.

Háromféle célfüggvénnyel, különböző időhorizonton végeztünk szabályozási számításokat. Az OPTNL segítségével 6, 9 és 10, a CONOPT-tal pedig 4, 6 és 9 időszakra vonatkozóan kíséreltük meg a modell szabályozását. Mivel mindkét programmal ugyanazt a feladatot akartuk megoldani, kvadratikus célfüggvények alkalmazására szorítkoztunk, tehát nem használtuk ki azt a lehetőséget, amelyet a CONOPT esetén a gyakorlatilag tetszőleges célfüggvény alkal-

mazása jelent; ezen túlmenően, a jobb összehasonlíthatóság érdekében, az OPTNL-nél lemondunk a véletlen hatások vizsgálatáról. A célfüggvények háromféle gazdasági alternatívát képviseltek; mind a háromban közös elem volt a nem rubel elszámolású külkereskedelmi mérleg megfelelő aktívumának biztosítása. Ezen túlmenően az egyik változat elsősorban a termelés fejlesztését, a második elsősorban a fogyasztás fejlesztését, míg a harmadik a termelés, a fogyasztás és a beruházások egymással arányos fejlesztését tűzte ki célul. A vizsgálat arra irányult, hogy hogyan függ a magyar gazdaság optimálisnak tekintett növekedési pályája a szabályozás időhorizontjának hosszától, illetve a CONOPT és az OPTNL alapjául szolgáló különböző matematikai módszerektől. Számítási eredményeinket közgazdasági szempontból is jól lehetett interpretálni, a következőkben azonban, főként terjedelmi okokból, csupán a programok működésével kapcsolatos tapasztalatokról lesz szó.

A CONOPT-tal való szabályozáshoz 4 időszak esetén 409 CPU másodpercre, 6 időszak esetén 1209 CPU másodpercre volt szükség,<sup>6</sup> 9 időszak esetén a program közel degenerált megengedett megoldáshoz jutott, és erről a holtpontról nem sikerült kimozdítanunk, elsősorban azért, mert a szabályozási számításonkkal különböző — de nem módszertani jellegű — okok miatt egyelőre le kellett állnunk. Az OPTNL alkalmazásánál a felhasználó nem értesül az elhasznált CPU-időről, tehát ilyen jellegű összehasonlításra nem volt mód a kétféle programnál, az „összes” elhasznált gépidő azonos, ill. hasonló nagyságú feladatokról viszont hozzávetőleg megegyezett a két programra vonatkozóan (emlékeztetünk arra, hogy a modell minden esetben ugyanaz volt). Ehhez az összehasonlításhoz azt is hozzá kell tenni, hogy az OPTNL esetén a konvergencia nagyon lassú volt, és a számítást a pontos optimum megtalálása előtt le kellett állítanunk, gyakorlatilag minden esetben; az így kapott „közel optimális” megoldások viszont, mint előbb említettük, minden további nélkül alkalmasak voltak a gazdasági elemzés céljára.

A kétféle programmal meghatározott megoldások igen jól közelítették egymást, az eltérés lényegesen kisebb volt, mint a célfüggvénynek az optimális pályára gyakorolt hatása. A választott célfüggvények mellett a gazdasági növekedés pályájára az időhorizont hossza erősebb hatást gyakorolt, mint a célfüggvény.

Ezzel az OPTNL és a CONOPT első hazai alkalmazásaival kapcsolatos leglényegesebb tapasztalatainkat el is mondtuk. Jogos lehet az az igény, hogy további kísérleti számításokra, tapasztalatokra lenne szükség, legalábbis ami a magyar modellekhez kapcsolódó alkalmazásokat illeti, mivel — mint láttuk — külföldi alkalmazásokban és az ezekre vonatkozó tapasztalatokban nincs hiány. Mindent figyelembe véve, megítélésünk szerint már az eddigi hazai tapasztalatok birtokában is kijelenthetjük, hogy mind a két szóban forgó programcsomag a magyarországi gazdasági tervezés hasznos segédeszköze lehet. A két program összehasonlításával kapcsolatban még egy megjegyzés kívánkozik ide, ez a következő.

Az 1. és 2. fejezetben mondottak alapján, kivált a matematikai programozás művelői számára az a következtetés a kézenfekvő, hogy a CONOPT lényegesen jobb kód, mint az OPTNL. Ezzel a véleménnyel lényegében véve egyet lehet érteni — hiszen a CONOPT sokkal többet tud, mint az OPTNL: tetszőleges célfüggvény mellett lehet alkalmazni, egyenlőtlenségeket lehet vele kezelni

<sup>6</sup> Mint említettük, a számításokat az MTA SZTAKI IBM 3031 számítógépen végeztük.

stb. —, ez a különbség azonban az általunk végzett számítások során nem volt szembeszökő. Ennek az az oka, hogy bizonyos esetekben egy adott feladatot egy „jó” algoritmus kevésbé hatékonyan old meg, mint egy egyszerűbb, és ennél fogva kevésbé jónak tekintett eljárás. Ézzel kapcsolatban megemlítjük, hogy a 31 egyenletről álló modellünkben [16] eredetileg semmilyen egyéni korlát nem szerepelt a változókra vonatkozóan, mivel akkor egyértelműen csak az OPTNL alkalmazására gondoltunk. Mint arra előbb már utaltunk, a számítás jól sikerült. Abban az időben a CONOPT-program még egyetlen hazai számítóközpontban sem állt rendelkezésre, lehetőségünk nyílt azonban arra, hogy a szóban forgó modellt kísérleti célra kiküldjük Washingtonba. A CONOPT-programmal való ottani futtatást személyesen A. Drud irányította. Mint később megtudtuk, az első kísérlet sikertelen volt, mivel — korlátok hiányában — az optimumkeresés során az egyik változó nulla értéket vett fel, és a számítás nullával való osztás miatt megállt. A sikeres futtatást ezután megfelelő alsó korlát beépítésével lehetett biztosítani.

(Beérkezett: 1986. március 12-én.)

#### IRODALOM

1. ABADIE, J.—J. CARPENTIER: Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints. In: R. Fletcher (ed.): *Optimization*, Academic Press, New York, 1969, 37—47.
2. BRADA, J.—A. KING—D. SCHLAGENHAUF: The optimality of socialist development strategies: An empirical inquiry. *Journal of Economic Dynamics and Control* 2 (1981) 1—27.
3. BROYDEN, C. G.: The convergence of a class of doublerank minimization algorithms. *Journal of the Institute of Mathematics and its Applications*, 6 (1970) 76—90.
4. CAUCHOIS, T.: The World Coffee Model. *Disszertáció*, M. I. T., 1980.
5. CHOW, G. C.: *Analysis and control of dynamic economic systems*. Wiley, New York, 1975.
6. CHOW, G. C.: *Econometric analysis by control methods*. Wiley, New York, 1981.
7. DRUD, A.—A. MEERAUS: *CONOPT — A system for large scale dynamic optimization — User's Guide*. Development Research Center, The World Bank, 1980.
8. DRUD, A.: *CONOPT — A GRG-code for large sparse dynamic nonlinear optimization problems*. Development Research Department, The World Bank, 1983.
9. FAURE, P.—P. HUARD: Résolution des programmes mathématiques à fonction non-linéaire par le méthode du gradient réduit. *Revue Française de Recherche Opérationnelle*, 9 (1965) 167—205.
10. FISCHER, J.-G. UEBE: The “optimal” control of the RWI Model. In: J. GRUBER (ed.): *Econometric decision models*. Springer, 1983.
11. FLETCHER, R.: A new approach to variable metric algorithms. *Computer Journal*, 13 (1970) 317—322.
12. FÖRMMHOLTZ, H.-J. WOLTERS: A control-theoretic analysis for a small econometric model of the Federal Republic of Germany. In: J. GRUBER (ed.): *Econometric decision models*. Springer, 1983.
13. GOLDFARB, D.: A family of variable metric updates, derived by variational means. *Mathematics of Computation*, 24 (1970) 23—26.
14. GREEN, D. W.—C. I. HIGGINS: *A macroeconomic model of the Soviet Union*. Academic Press, New York, 1977.
15. HELLERMAN, E.—D. RARICK: Reinversion with the preassigned pivot procedure. *Mathematical Programming*, 1 (1971) 195—216.
16. Kísérlet a H-4/B/1—83 egyszektoros ökonometriai modell szabályozására. *Munkaanyagok* 8., MTA Közgazdaságtudományi Intézet, Ökonometriai Laboratórium, Budapest, 1983. november.
17. KÖRNAI J.—MÁRTOS B. (szerk.): *Szabályozás árjelzések nélkül*. Akadémiai Kiadó, Budapest, 1981.
18. LIGETI I.—GYURKOVICS É.—KLEMENCICS M.—VÖRÖS T.: A makrostruktúra és a

nem rubel külkereskedelmi mérleg. Országos Tervhivatal Tervgazdasági Intézet, Budapest, 1985.

19. LIGETI I.—SIVÁK J.: *Növekedés, szabályozás és stabilitás a gazdasági folyamatokban*. Közgazdasági és Jogi Könyvkiadó, Budapest, 1978.
20. NECK, R.—U. POSCH: On the "optimality" of macroeconomic policies: an application to Austria. In: G. Feichtinger (ed.): *Optimal control theory and economic applications*. North Holland, Amsterdam, 1982.
21. SANDBLOM, C. L.—J. BANASIK: Economic policy with bounded controls. *Economic Modelling* 2 (1985) 135—148.
22. SCHITTKOWSKI, K.: *Nonlinear programming codes — Information, tests, performance*. Springer, Berlin—Heidelberg—New York, 1980.
23. SHANNO, D. F.: Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, 24 (1970) 647—656.
24. WOLFE, P.: Methods for linear constraints. In: *Nonlinear programming* (ed. J. Abadie et al.) North-Holland Publishing Company, Amsterdam, 1967.