Simon György

A reflektorprogramozás elvei és algoritmusa

1. Bevezetés

Az első kérdés, amelyre válaszolnunk kell: *mi a reflektorprogramozás?* Tágabb értelemben így nevezhetjük mindazokat a lehetséges programozási eljárásokat, amelyek felhasználják az ún. reflektorelvet (lásd a cikk 2. részét). Szűkebb értelemben a reflektorprogramozás jelenleg heurisztikus alapokon nyugvó algoritmus nagyméretű lineáris programozási feladatok megoldására. Az utóbbi értelemben használjuk a kifejezést a cikk további részében, ahol kísérletet teszünk a reflektorprogramozási modellek és algoritmus ismertetésére. A konkrét eljárás használhatóságát kísérleti számítások verifikálják, a konvergencia matematikailag még nincs bizonyítva.

A reflektorprogramozással kapcsolatos kutatásokra a szerzőt azok a problémák ösztönözték, amelyek a dekompozíciós eljárásoknak (lásd elsősorban [1]-et és [2]-t) a népgazdaságtervezésben történő alkalmazása kapcsán merültek fel. A nehézségek (lassú vagy nem kielégítő konvergencia, a központi feladat összeállításának és megoldásának problémái stb.) elemzése arra a következtetésre vezetett, hogy nagyméretű feladatok megoldására szolgáló hatékony programozási eljárás nem ignorálhatja a „szektorok" közötti, *horizontális* kapcsolatokat. Egy ilyen módosítás azonban a dekompozíciós szemlélet gyökeres felülvizsgálását, a dekompozíciós elvnek a reflektorelvvel való helyettesítését követelte meg. Utóbbival függ össze a reflektorprogramozás elnevezése, amelynek első ismertetését lásd [3]-ban, továbbá [4]-ben és [5]-ben.

Az utóbbi években kísérleti számításokat folytattunk. A számítások első szakaszának eredményeit a [7] tanulmány foglalta össze. Jelen előadás a reflektorprogramozásnak a kísérleti számítások tapasztalatai figyelembevételével továbbfejlesztett algoritmusát ismerteti.

A reflektorprogramozás alapján bizonyos következtetések vonhatók le az ésszerű gazdaságirányítási rendszer sajátosságaival kapcsolatban. Ezzel a kérdéssel az [5] és [6] munka foglalkozott részletesebben.

2. A reflektorprogramozás elvei

A reflektorprogramozás néhány, egymással kapcsolatban álló elvre épül. Három főbb elvet emelünk ki: 1. reflektorelv, 2. policentrikusság elve, 3. sztereoszkopikus látás elve.

A *reflektorelv*et analógia alapján érthetjük meg. A fényszórót bekapcsolva a megvilágítás erőssége szempontjából több terület keletkezik. A legerősebb megvilágítást a reflektor fókuszában kapjuk, de létrejönnek közepesen és gyengén megvilágított területek is, miközben a fényszóró kezelőjének (pl. autóvezetőnek) utóbbiak is fontos, gyakran nélkülözhetetlen információt nyújtanak. Tételezzük fel, hogy valamely bonyolult feladat elvégzéséhez nagy területet kell erősen megvilágítani. Ugyanakkor olyan reflektorokkal rendelkezünk, amelyek csak viszonylag kis területet képesek erősen (fókuszált fénnyel) megvilágítani. Ezért *úgy állítjuk be a reflektorokat, hogy együttesen biztosítsák a nagy terület erős megvilágítását*, miközben minden egyes reflektor csak viszonylag kis területre bocsát fókuszált fényt, bár gyengén az egész területet láthatóvá teszi.

Ezzel a feladatot meg is oldottuk volna, ha minden részvevő látná valamennyi reflektor fényét. Esetünkben azonban más a helyzet. A reflektorok különböző színű fényt bocsátanak ki, s a részvevők (a reflektorok és részvevők száma azonos) e színárnyalatok közül csak egyet, mindegyik másikat képes *közvetlenül* érzékelni, mivel speciális szemüveget viselnek, amely csak bizonyos színű fényt bocsát át.

Feltételezzük, hogy minden részvevő azon a területen tevékenykedik, amelyet saját reflektora világít meg erősen. Ahhoz azonban, hogy feladatát sikeresen elvégezze, lényegesen több információval kell rendelkezni a *többi* területről, mint amit saját reflektorának gyenge fénye biztosít. Ezért a részvevők információs kapcsolatba lépnek egymással: mindegyik közli a saját területére vonatkozó leglényegesebb, a *többiek számára is fontos* ismereteit az összes többi részvevővel.

Gyakorlatilag az információáramlás információs központ vagy központok közbeiktatásával egyszerűsíthető, ahol bármely részvevő csupán egy helyre ad át (ill. egy helyről kap) információt.

Tételezzük fel, hogy *minden részvevő a többiektől kapott információ alapján megváltoztatja tervezett tevékenységét, s erről informálja a többi részvevőt,* utóbbiak szintén módosítják magatartásukat stb. Ily módon egy hosszabb, soklépéses, iteratív folyamathoz jutunk. Ebben jelölhető meg a fentebb vázolt *reflektorelv dinamikus aspektusa.* Nem nehéz belátni, hogy a reflektorelv tulajdonképpen az *általános visszacsatolás elvét* is tartalmazza.

Mindez természetesen önmagában nem elegendő ahhoz, hogy a részvevők bizonyos számú iteráció után eljussanak a bonyolult feladat legjobb megoldásához, a globális optimumhoz, vagy legalábbis annak közelébe. Az eljárás konvergenciája azon múlik, hogy

a) miként használják az egyes részvevők saját reflektoraikat (hogyan állítják össze reflektorprogramozási feladataikat);

b) milyen információt adnak át a többieknek és hogyan építik be a többiektől kapott információt saját reflektorprogramozási modelljeikbe.

E kérdésekről később lesz szó, a lineáris programozás általános feladata vonatkozásában. Előbb a reflektorprogramozás két további elvét ismertetjük.

A *policentrikusság elve* azt fejezi ki, hogy a reflektorprogramozás nem tételez fel valamely, a szektoroktól elkülönült központot: *minden szektor a maga területén központ*ként funkcionál. A policentrikusság elve lényegében a reflektorelvből következik, hasonlóan ahhoz, ahogy a dekompozíciós elvből az egyközpontúság és hierarchikus többszintűség.

Úgy vélem, hogy a policentrikusság pontosabban megfelel a gazdasági valóságnak, mint például a DW eljárás által feltételezett egyközpontúság. Így például *a KGST-ben részvevő országok közötti tervkoordináció csak a policentrikusság figyelembevételével modellezhető.* De sok tekintetben hasonló a helyzet az egyes szocialista országokon belül is, különösen a gazdasági reformok kibontakozása óta.

Látni kell továbbá, hogy a policentrikus irányítás határesetként az egyközpontú irányítást is tartalmazza, továbbá a policentrikus rendszerben résztvehetnek, a reflektorprogramozással modellezhetők a szektoroknál (ágazatoknál vagy vállalatoknál) magasabb szintű gazdasági irányítószervek (minisztériumok, területi szervek, Tervhivatal stb.) is. Ebben az értelemben a reflektorprogramozás a *vertikális és horizontális kapcsolatok együttes figyelembevételével* történhet.

A reflektorprogramozás tervezési alkalmazásának egyik potenciális előnye, hogy egyetlen, *időben párhuzamosan* futó folyamattá integrálja a különböző szinteken (vállalatoknál, minisztériumokban, Tervhivatalban stb.) folyó tervezést, ahol minden szerv minden más szervvel állandó információs kapcsolatban, kölcsönhatásban áll, s ily módon kikapcsolja, minimálisra csökkenti a tervezés ,,holtidőit''.

A *sztereoszkopikus látás elvét* a reflektorelv kapcsán tárgyalt analógiából kiindulva értelmezhetjük. Tételezzük fel, hogy minden reflektor kettős fényt bocsát ki, s a részvevők speciális szemüvegei átbocsátják saját reflektoruk kettős fényét: egyiket a jobb, másikat a bal szemhez, ily módon sztereoszkopikus látást biztosítva.

A reflektor kettős fényének esetünkben a matematikai programozás primál és duál feladata felel meg. Másképp fogalmazva: a reflektorprogramozás egyidejűleg támaszkodik volumen- és árinformációkra a globális optimum közelítése során. Vagyis a sztereoszkopikus látás elve esetünkben megfelel a *primál és duál feladatok kombinálása elvének*.

A primál és duál feladatok eredményeit *bizonyos mértékben* a DW eljárás, valamint a ,,kétszintű tervezés'' is kombinálják. A központ itt árakat *vagy* korlátokat ír elő a szektoroknak, a szektorok volumeneket *vagy* árakat jelentenek a központnak. A reflektorprogramozásban azonban a vertikális információáramlás helyett (vagy mellett) megjelennek a horizontális (szektorok közötti) volumen- és árinformációk, *egyszerre* adják át és használják fel *mindkétféle* információt. Ebben az értelemben a sztereoszkopikus látás elve a reflektorprogramozás sajátossága.

Úgy gondolom, hogy a vázolt elvek érvényesülése módot nyújt a reflektorprogramozás felhasználására nemcsupán a hagyományos központi tervezés keretében, hanem *a szocialista tervpiac integrált szimulálására* is, mind egy-egy ország, mind pedig nagyobb gazdasági közösségek (pl. KGST) keretében. A reflektorprogramozás alkalmazásának egyidejűleg, egymással kombináltan adható tervezési, piaci és szimulációs aspektus. Mit jelent ez kissé közelebbről?

A reflektorprogramozás *tervezési aspektusa* kifejeződhet a központi tervező szervek, vertikális összefüggések bekapcsolásában, volumeninformációk szisztematikus felhasználásában, társadalmi célfüggvény figyelembevételében.

Az eljárás *piaci aspektusa* megnyilvánulhat a részvevők viszonylagos önállóságában (policentrikusság), a horizontális, piaci kapcsolatok figyelembevételében, az árak és hozzájuk kapcsolódó gazdasági ösztönzők bekapcsolásában.

A *szimulációs aspektus* a folyamat iteratív jellegében, valamint abban fejeződhet ki, hogy a részvevők a kapott eredmények függvényében módosíthatják saját területük kiinduló adatait, miután adott kiinduló adatokkal az eljárás már eljutott a globális optimumhoz, ill. annak közelébe.

E kérdések részletes kifejtése azonban túlmegy jelen munka tárgyán (lásd [5], [6]). A továbbiakban a szűkebb értelemben vett reflektorprogramozás algoritmusáról lesz szó.

3. A reflektorprogramozási modellek szerkezete összeállítási szabályai

A reflektorprogramozás alapfeladata azonos a lineáris programozás általános (felső korlátokat, egyenlőségeket és alsó korlátokat tartalmazó) feladatával. A feltételrendszer alakjára semmiféle megszorítást nem teszünk. Az eljárást a lineáris programozás ún. normálfeladatán mutatjuk be, tekintettel arra, hogy minden lineáris programozási feladat visszavezethető a normálfeladatra, két normálfeladat egymás utáni megoldása révén [8].

Az alapfeladatot tehát a lineáris programozás szokásos primál—duál feladatpárjával jellemezhetjük.

Primál feladat *Duál feladat*

$$(3.1)\quad \begin{cases} a)\ Ax \leq b \\ b)\ cx \to \max \\ c)\ \quad x \geq 0 \end{cases} \qquad (3.2)\quad \begin{cases} a)\ pA \geq c \\ b)\ pb \to \min \\ c)\ p \quad \geq 0 \end{cases}$$

Az egyszerűbb tárgyalás kedvéért a (3.1) primál feladatot a volumentervezési feladattal (x vektor komponensei a tevékenységek volumenei), a (3.2) duál feladatot pedig az ártervezési feladattal (p vektor komponensei az árak) azonosítjuk. A latin nagybetűk matrixot, a latin kisbetűk vektort jelölnek. A sor- és oszlopvektort helye különbözteti meg egymástól. Az A matrix és c vektor tartalmazhat negatív elemeket is, a b vektor nem negatív.

A leírás egyszerűsítése érdekében a továbbiakban a primál feladattal foglalkozunk, mivel annak optimális megoldása szimplex módszerrel megadja nemcsak az optimális tevékenységvolumeneket (x^0), hanem az optimális árakat (p^0), más néven árnyékárakat is.

A reflektorprogramozás modellrendszerét a következő főbb szabályok szerint alakítjuk ki (jelölésük F1, F2 stb.):

F1. Minden primál és duál változó az alapfeladatnak megfelelő részletezettséggel (ún. saját változóként) kell hogy szerepeljen *legalább egy* reflektorprogramozási feladatban.

F2. Minden reflektorprogramozási modell fel kell hogy ölelje a megfelelő (primál vagy duál) alapfeladatot, de a *nem saját* változók tekintetében transzformált (sűrített) alakban.

F3. A *nem saját primál és duál változók* (külső tevékenységek és külső korlátozó feltételek) *összevonásá*hoz azokat az értékeket kell felhasználni, amelyek e változókra olyan reflektorprogramozási feladatok optimális megoldásaként adódtak, ahol a szóban forgó változók saját változóként szerepeltek. Ha valamely változó egynél több reflektorprogramozási modell saját változója, a számtani átlagérték alkalmazandó.

F4. Az *első iterációban* az összevonáshoz a priori értékeket kell alkalmazni, amelyek lehetnek pl. bázisidőszaki volumen- és áradatok, prognosztizált adatok vagy — triviális esetben — nullvektorok. A további iterációkban mindig a megelőző iteráció eredményei használandók fel a nem saját változók összevonásához.

F5. Az alapfeladat transzformált alakján kívül a reflektorprogramozási feladatok ún. *árstabilizáló faktorokat*, valamint *volumenmódosító alsó korlátokat* is kell hogy tartalmazzanak, ha a primál feladatban vannak lokális korlátozó feltételek. *Lokális korlátozó feltételnek* nevezzük azokat a feltételeket, amelyek csak egy feladat saját primál változóit korlátozzák. A többi korlátozó feltétel *nem-lokális.*

F6. A *nem saját primál változók* (külső tevékenységek) minden reflektorprogramozási feladatban két változóvá vonandók össze:

1. Aggregált *versenyző tevékenységek* (az adott reflektorprogramozási modell saját erőforrásait kibocsátó nem saját tevékenységek);
2. *Egyéb nem saját tevékenységek.*

F7. A *nem saját duál változók* (külső korlátozó feltételek) az alábbi négy korlátozó feltétellé vonandók össze:

1. Aggregált *nemlokális termékfeltételek,* más szóval negatív elemet tartalmazó, vagyis olyan feltételek, amelyek a külső tevékenységek által kibocsátott erőforrásokra vonatkoznak;
2. Aggregált *nemlokális egyéb feltételek;*
3. Aggregált versenyző tevékenységek lokális feltételei (összevontan);
4. Aggregált egyéb nem saját tevékenységek lokális feltételei (összevontan).

A *reflektorprogramozási primál feladat általános alakját a* (3.3)—(3.7) *összefüggések tartalmazzák,* egyúttal formalizálva a reflektorprogramozás lineáris esetre kidolgozott algoritmusának néhány alapvető elemét.

Az alábbiakban megadjuk a (3.3)—(3.7) összefüggésekben szereplő jelölések magyarázatát.

Jobb alsó indexek:

t = iterációs lépés sorszáma (pl. $t = 1$ az első iterációt jelöli);
s = reflektorprogramozási modell sorszáma.

Matrixok esetében két s áll egymás mellett, az első a sorokra, a második az oszlopra vonatkozik. Aláhúzással jelöltük, ha a sorok vagy oszlopok a reflektorprogramozási modell *nem saját* változóihoz tartoznak: pl. A_{ss} matrixnak mind sorai, mind oszlopai az s feladat saját változóira vonatkoznak, ezzel szemben $A_{\underline{s}s}$ matrixnak csak az oszlopai. Analóg értelme van *vektorok s* indexének: pl. x_s vektor az s feladat saját primál változóit, $x_{\underline{s}}$ vektor nem saját primál változóit jelöli.

Bal felső indexek:

k = versenyző tevékenység sorszáma;
z = egyéb nem saját tevékenységek sorszáma;
q = nemlokális terméktípusú (nem saját) feltételek sorszáma;
r = nemlokális (nem saját) feltételek sorszáma.

Matrixok esetén a bal felső kettős indexekre is érvényes, hogy az első index a matrix soraira, a második oszlopaira vonatkozik. Egy bal felső matrixindex esetén az összefüggésből értelemszerűen adódik, hogy sorokról vagy oszlopokról van-e szó.

Reflektorprogramozási modell

(Primál feladat)

(3.3)

$$
\left\{
\begin{array}{lll}
\text{a)} & A_{ss}\, x_{st} + {}^{k}\gamma_{st}\, |\, {}^{k}A_{ss}\, {}^{k}x_{\underline{s}(t-1)}\, | + {}^{z}\gamma_{st}\, |\, {}^{z}A_{s\underline{s}}\, {}^{z}x_{\underline{s}(t-1)}\, | & \leq b_{s} \\
\text{b)} & |\, {}^{q}p_{\underline{s}(t-1)}\, {}^{q}A_{s\underline{s}}\, |\, x_{st} + {}^{k}\gamma_{st}\, |\, {}^{q}p_{\underline{s}(t-1)}\, {}^{qk}A_{\underline{ss}}\, {}^{k}x_{\underline{s}(t-1)}\, | + {}^{z}\gamma_{st}\, |\, {}^{q}p_{s(t-1)}\, {}^{qz}A_{\underline{ss}}\, {}^{z}x_{\underline{s}(t-1)} & \leq {}^{q}p_{\underline{s}(t-1)}\, {}^{q}b_{\underline{s}} \\
\text{c)} & |\, {}^{r}p_{s(t-1)}\, {}^{r}A_{s\underline{s}}\, |\, x_{st} + {}^{k}\gamma_{st}\, |\, {}^{r}p_{\underline{s}(t-1)}\, {}^{rk}A_{\underline{ss}}\, {}^{k}x_{\underline{s}(t-1)}\, | + {}^{z}\gamma_{st}\, |\, {}^{r}p_{\underline{s}(t-1)}\, {}^{rz}A_{\underline{ss}}\, {}^{z}x_{\underline{s}(t-1)}\, | & \leq {}^{r}p_{\underline{s}(t-1)}\, {}^{r}b_{\underline{s}} \\
\text{d)} & {}^{k}\gamma_{st}\, |\, {}^{k}p_{\underline{s}(t-1)}\, {}^{kk}A_{\underline{ss}}\, {}^{k}x_{\underline{s}(t-1)}\, | \qquad + {}^{k}u_{st} & = {}^{k}p_{\underline{s}(t-1)}\, {}^{k}b_{\underline{s}} \\
\text{e)} & {}^{z}\gamma_{st}\, |\, {}^{z}p_{\underline{s}(t-1)}\, {}^{zz}A_{\underline{ss}}\, {}^{z}x_{\underline{s}(t-1)}\, | \qquad + {}^{z}u_{st} & = {}^{z}p_{\underline{s}(t-1)}\, {}^{z}b_{\underline{s}} \\
\text{f)} & {}^{k}\gamma_{st}\, |\, {}^{k}p_{\underline{s}(t-1)}\, {}^{kk}A_{\underline{ss}}\, {}^{k}x_{\underline{s}(t-1)}\, | \qquad - {}^{k}v_{st} & = \alpha_{st}\, |\, {}^{k}p_{\underline{s}(t-1)}\, {}^{k}b_{\underline{s}}\, | \\
\text{g)} & {}^{z}\gamma_{st}\, |\, {}^{z}p_{\underline{s}(t-1)}\, {}^{zz}A_{\underline{ss}}\, {}^{z}x_{\underline{s}(t-1)}\, | \qquad - {}^{z}v_{st} & = \alpha_{st}\, |\, {}^{z}p_{\underline{s}(t-1)}\, {}^{z}b_{\underline{s}}\, | \\
\text{h)} & c_{s}\, x_{st} + {}^{k}\gamma_{st}\, |\, {}^{k}c_{\underline{s}}\, {}^{k}x_{\underline{s}(t-1)}\, | + {}^{z}\gamma_{st}\, |\, {}^{z}c_{\underline{s}}\, {}^{z}x_{\underline{s}(t-1)}\, | + {}^{k}\beta_{st}\, {}^{k}u_{st} + {}^{z}\beta_{st}\, {}^{z}u_{st} \to \max & \\
\text{i)} & x_{st},\, {}^{k}\gamma_{st},\, {}^{z}\gamma_{st},\, {}^{k}u_{st},\, {}^{z}u_{st},\, {}^{k}v_{st},\, {}^{z}v_{st} \geq 0 &
\end{array}
\right.
$$

(3.4) $$\alpha_{st} = \gamma_{s(t-1)} \min + \varepsilon \mid 1 - \gamma_{s(t-1)} \min \mid$$

ha $t > t_1$ (t_1 = első programozási szakasz iterációinak száma)
ha $t \leq t_1$ akkor $\alpha_{st} = 0$

(3.5) $$\gamma_{s(t-1)} \min = \min \mid {}^k\gamma_{s(t-1)}, {}^z\gamma_{s(t-1)} \mid$$

Megjegyzés: α_{st} értéke mindaddig nem változtatandó, amíg adott α_{st}-vel két egymást követő iterációban számottevő (pl. 5%-nál nagyobb) változás megy végbe a saját változók értékében.

(3.6) $$0 < \varepsilon \ll 1$$

(3.7.1) $${}^k\beta_{st} = \sqrt{\frac{{}^k\beta_{s(t-1)} \mid {}^k\underline{p}_{s(t-1)} {}^k\underline{b}_s \mid}{{}^k\underline{p}_{st} {}^k\underline{b}_s}};$$

(3.7.2) $${}^z\beta_{st} = \sqrt{\frac{{}^z\beta_{s(t-1)} \mid {}^z\underline{p}_{s(t-1)} {}^z\underline{b}_s \mid}{{}^z\underline{p}_{st} {}^z\underline{b}_s}}$$

(3.7.1), (3.7.2) alkalmazandó, ha $t > 1$; $t = 1$ esetén $0 < {}^k\beta_{st}, {}^z\beta_{st} \leq 1$ (3.7.3)

A (3.3) összefüggésben ${}^k\gamma_{st}$ skalár az aggregált versenyző tevékenységek terjedelme, ${}^z\gamma_{st}$ skalár pedig az egyéb aggregált tevékenységé.

A (3.3d) és (3.3e) feltételek eredetileg felső korlátok. A maradékváltozók (${}^k u_{st}$, ${}^z u_{st}$ egyelemű vektorok) felhasználásával egyenlőségként írtuk fel őket, mivel maradékváltozóikhoz nem zéró célfüggvénykoefficiensek (${}^k\beta_{st}$, ${}^z\beta_{st}$) tartoznak.

A (3.3f), (3.3g) feltételek *volumenmódosító alsó korlátok*, a szokásos módon (lásd pl. [8]-at) egyenlőséggé átalakítva. A feleslegváltozókat ${}^k v_{st}$, ill. ${}^z v_{st}$ (mindkettő egyelemű vektor) jelöli. Az α_{st} skalárt, melynek képzési szabályait a (3.4)—(3.6) összefüggések adják meg, *volumenmódosító faktor*nak nevezzük.

A (3.4) összefüggés kapcsán szereplő *programozási szakasz* kifejezéshez további magyarázatként megjegyezzük: az *első programozási szakasz*t ($t < t_1$) az jellemzi, hogy a volumenmódosító faktor (α_{st}) zéró értékkel szerepel az összefüggésekben, vagyis e szakaszban a (3.3f), (3.3g) feltételek gyakorlatilag ki vannak iktatva a reflektorprogramozási feladatokból. Az első programozási szakasz addig tart, amíg a saját változók értékének változása két egymást követő iterációban *mindenütt* egy bizonyos előre megadott minimális érték (pl. 5%) alá csökken. Ez az ún. *leállási feltétel.*

A második szakasz elején α_{st} pozitívvá válik, a (3.4)—(3.6) képletnek megfelelően (pl. $\varepsilon = 0{,}20$), s ismét végrehajtandó annyi iteráció (változatlan α_{st}-vel), amennyi a leállási feltétel teljesítéséhez kell. Ezután módosítandó α_{st} értéke, s kezdődik a harmadik programozási szakasz stb. Adott (pl. 5%-os) leállási feltétellel a reflektorprogramozás néhány szakasz után automatikusan befejeződik.

Az *árstabilizáló faktorok*at összefüggéseinkben a ${}^k\beta_{st}$, illetve ${}^z\beta_{st}$ skalár jelöli. A (3.7.1)—(3.7.3) összefüggések megadják az árstabilizáló faktorok képzési szabályait. Kísérleti számításaink során az árstabilizáló faktorok kezdő értékét ($t = 1$) egységesen 0,3-nak választottuk.

A formális összefüggések bemutatása után az F1 szabály kiegészítéseként *pótlólagos szabályok*at (jelölésük P1, P2 stb.) ismertetünk a reflektorprogramozási feladatok saját változói (primál és duál) nómenklatúrájának meghatározását illetően.

P1 A saját változók kijelölését a *primál változók*kal kezdjük, utóbbiakat annyi, kb. *egyenlő nagyságú* csoportra osztva, hogy az adott számítókapacitással könnyen kezelhető méretű programozási feladatokhoz jussunk.

P2 Az ún. *indikátorváltozók*, illetve célfüggvényváltozók („kantorovicsi") típusú célfüggvény esetén) minden reflektorprogramozási modellben saját változóként kezelendők.

P3 Minden *lokális korlátozó feltétel* annak a modellnek a saját feltétele, amelynek saját primál változóit korlátozza.

P4 *Specializált nemlokális termékfeltételek* (olyan nemlokális termékfeltételek, amelyeknek egy feladat saját primál változói oszlopában van negatív elemük) ott emelendők ki, ahol negatív elemük van.

P5 *Despecializált nemlokális termékfeltételek* (több feladat saját primál változóinak oszlopában van negatív elemük) mindenütt saját feltételként kezelendők, ahol negatív elemük van.

P6 A *nemlokális egyéb feltételek* mindig egy olyan feladatban emelendők ki, ahol az A matrix megfelelő sorelemeinek összege (a feladat saját primál változóit figyelembevéve) nemzéró, s ugyanakkor az addig kijelölt saját nemlokális egyéb feltételek száma legkisebb. Több legkisebb érték esetén abban a feladatban, ahol az A matrix megfelelő sorelemeinek összege legnagyobb.

Ezzel áttekintettük a reflektorprogramozási modellek szerkezetét, összeállítási szabályait. A továbbiakban, folytatva a reflektorprogramozás algoritmusának leírását, az információs kapcsolatokkal, majd pedig a konvergenciakritériumokkal foglalkozunk.

4. Információs kapcsolatok

A tárgyalás megkönnyítése érdekében mindenekelőtt néhány új jelölést ismertetünk.

$$A_{ss} = \begin{bmatrix} {}^{q}A_{ss} \\ \hdashline {}^{r}A_{ss} \\ \hdashline {}^{h}A_{ss} \end{bmatrix}, \text{ továbbá } {}^{\underline{h}}A_{ss} = \begin{bmatrix} {}^{q}A_{ss} \\ \hdashline {}^{r}A_{ss} \end{bmatrix} \tag{4.1}$$

Itt ${}^{h}A_{ss}$ az A_{ss} matrix lokális saját erőforrásokhoz, ${}^{\underline{h}}A_{ss}$ pedig nemlokális saját erőforrásokhoz tartozó sorait tartalmazza. ${}^{q}A_{ss}$ az A_{ss} matrix terméktípusú, ${}^{r}A_{ss}$ egyéb nemlokális saját erőforrásaihoz tartozó sorokból áll. Továbbá bevezetjük az alábbi jelölést:

$${}^{\underline{h}}A_{ss} = \begin{bmatrix} {}^{q}A_{ss} \\ \hdashline {}^{r}A_{ss} \end{bmatrix}. \tag{4.2}$$

Vagyis ${}^{\underline{h}}A_{ss}$ bármely eleme az s feladat saját primál változóinak fajlagos igénye nemlokális (terméktípusú és egyéb) külső erőforrásokból.

Analóg módon felbontjuk a p_{st}^{0} árvektort is (a ^{0} jel itt a reflektorprogramozási feladat szerinti optimumot jelöli), valamint a b_s vektort.

$$p_{st}^{0} = [{}^{q}p_{st}^{0} \,\vdots\, {}^{r}p_{st}^{0} \,\vdots\, {}^{h}p_{st}^{0}], \text{ továbbá } {}^{\underline{h}}p_{st}^{0} = [{}^{q}p_{st}^{0} \,\vdots\, {}^{r}p_{st}^{0}] \tag{4.3}$$

${}^{h}p^0_{st}$ a lokális saját erőforrások árvektora, ${}^{\underline{h}}p^0_{st}$ pedig a nemlokális saját erőforrásoké.

(4.4) $$b_s = \begin{bmatrix} {}^{q}b_s \\ \hline {}^{r}b_s \\ \hline {}^{h}b_s \end{bmatrix}, \text{ továbbá } {}^{\underline{h}}b_s = \begin{bmatrix} {}^{q}b_s \\ \hline {}^{r}b_s \end{bmatrix}.$$

Itt ${}^{h}b_s$ vektor elemei a lokális saját erőforrások, ${}^{\underline{h}}b_s$ vektoré pedig a nemlokális saját erőforrások rendelkezésére álló mennyiségeit tartalmazza.

Az s reflektorprogramozási feladat a többi feladat részére általában az alábbi információt kell hogy eljuttassa (a $t+1$ iteráció előtt).

Vektorok:

(4.5) ${}^{\underline{h}}A_{\underline{ss}}\, x^0_{st}$ = Külső erőforrások iránti igény (természetben)

(4.6) ${}^{\underline{h}}p^0_{st}$ = Nemlokális saját erőforrások árai

Skalárok:

(4.7) ${}^{q}p^0_{st}\,{}^{q}A_{ss}\, x^0_{st}$ = Terméktípusú nemlokális saját erőforrások kibocsátásának (—) és felhasználásának (+) értékegyenlege (despecializált erőforrások kiemelendők)

(4.8) ${}^{r}p^0_{st}\,{}^{r}A_{ss}\, x^0_{st}$ = Egyéb nemlokális saját erőforrások együttes felhasználása értékben

(4.9) ${}^{h}p^0_{st}\,{}^{h}A_{ss}\, x^0_{st}$ = Lokális saját erőforrások kibocsátásának (—) és felhasználásának (+) értékegyenlege

(4.10) ${}^{q}p^0_{st}\,{}^{q}b_s$ = Terméktípusú nemlokális saját erőforrások értékösszege (despecializált erőforrások kiemelendők)

(4.11) ${}^{r}p^0_{st}\,{}^{r}b_s$ = Egyéb nemlokális saját erőforrások értékösszege

(4.12) ${}^{h}p^0_{st}\,{}^{h}b_s$ = Lokális saját erőforrások értékösszege

(4.13) $c_s\, x^0_{st}$ = Célfüggvényérték a saját blokkban (saját primál változók oszlopaiban)

Ha a saját tevékenységek (primál változók) között versenyző tevékenységek is vannak, ezek adatait kiemelten meg kell adni a megfelelő reflektorprogramozási modellek részére.

Az olyan primál változók adatait, amelyek más modellekben is saját változóként szerepelnek (pl. indikátor-változók), szintén kiemelten, a (4.5)—(4.9), valamint (4.13) összefüggésekkel analóg módon kell közölni a megfelelő modellek összeállítóival.

Nem nehéz megmutatni, hogy ennyi információ elegendő a reflektorprogramozási modellek folyamatos, bármely t iteráció utáni átdolgozásához. Ennek részleteire nem térünk ki.

5. Konvergenciakritériumok és számítási tapasztalatok

A reflektorprogramozás sajátosságaival függ össze, hogy az eljárás konvergenciája *több, alternatív kritérium* alapján is ellenőrizhető. *Közvetlen* konvergenciakritériumoknak fogjuk nevezni a saját változók, *közvetett*nek a nem saját változók értékeire kapott programozási eredményeket felhasználó kritériumokat. A közvetlen és közvetett konvergenciakritériumok kölcsönösen helyettesítik egymást, vagyis elegendő vagy csak a közvetlen, vagy csak a közvetett kritériumok alakulását kísérni figyelemmel.

Melyek a *közvetlen konvergenciakritériumok?*

1. A *saját blokkok együttes célfüggvényértékének növekedése.*

Képletben:

$$\sum_{s} c_s x_{st}^0 \to \max . \tag{5.1}$$

Ha valamely primál változó több reflektorprogramozási feladat saját változója (pl. indikátorváltozó), átlagértéke veendő figyelembe.

2. A *feltételi rendszer inkonzisztenciájának megszűnése.*

Képletben:

$$(p_t^0 A x_t^0 - p_t^0 b) \to 0 \tag{5.2}$$

ahol p_t^0, ill. x_t^0 vektor komponensei a t iteráció árnyékárai, illetve tevékenységvolumenei a reflektorprogramozási modellek saját változóinak figyelembevételével (szükség esetén az F3 szabály szerint átlagolva).

A *közvetett konvergenciakritériumok*at alább adjuk meg.

1. *A versenyző és egyéb nem saját tevékenységek terjedelmének kölcsönös és egyre teljesebb elismerése.* Képletben:

$$\left.\begin{cases} \text{a)} \ {}^{k}\gamma_{st} \to 1 \\ \text{b)} \ {}^{z}\gamma_{st} \to 1 \end{cases}\right\} \text{(minden } s\text{-re)} \tag{5.3}$$

Nem nehéz belátni, hogy (5.3) teljesülése esetén a *volumenmódosító faktornak* is 1-hez kell konvergálni. Képletben:

$$\alpha_{st} \to 1 \text{ (minden } s\text{-re)} \tag{5.4}$$

Továbbá zéróhoz kell konvergálni a (3.3) összefüggésben szereplő *maradék és felesleg változók*nak. Képletben:

$$\left.\begin{bmatrix} \text{a)} \ {}^{k}u_{st} \to 0 \\ \text{b)} \ {}^{z}u_{st} \to 0 \\ \text{c)} \ {}^{k}v_{st} \to 0 \\ \text{d)} \ {}^{z}v_{st} \to 0 \end{bmatrix}\right. \text{(minden } s\text{-re)} \tag{5.5}$$

2. Az *árstabilizáló faktorok 1-hez való konvergálása*, vagyis lényegében a lokális erőforrások árainak kölcsönös és egyre teljesebb elismerése. Képletben:

$$(5.6) \qquad \left.\begin{cases} \text{a)} & {}^{k}\beta_{st} \to 1 \\ \text{b)} & {}^{z}\beta_{st} \to 1 \end{cases}\right\} \text{(minden } s\text{-re)}$$

3. A *nemlokális erőforrások árainak kölcsönös és egyre teljesebb elismerése*. Képletben:

$$(5.7) \qquad \left.\begin{cases} \text{a)} & {}^{k}\delta_{st} \to 1 \\ \text{b)} & {}^{z}\delta_{st} \to 1 \end{cases}\right\} \text{(minden } s\text{-re)}$$

ahol ${}^{k}\delta_{st}$, ${}^{z}\delta_{st}$ az aggregált nemlokális termék, ill. egyéb erőforrások árnyékára az s feladat t iterációjában.

Összefoglalóan megjegyezzük, hogy (5.3) teljesülése elégséges feltétele a reflektorprogramozás konvergenciájának. A konvergenciakritériumok közül csupán (5.2) figyelemmel kísérése igényel számottevő pótlólagos munkát. Gyakorlatilag elegendő a szokásoshoz legközelebb álló (5.1), valamint a feladatok megoldásából további számítás nélkül adódó közvetett konvergenciakritériumok alakulását figyelni.

Mit mutatnak a *számítási tapasztalatok* a reflektorprogramozás konvergenciáját illetően? Elöljáróban megjegyezzük, hogy a számítások Gábor Győző és Mezei Mihály részvételével, a MAVEMI GIER típusú elektronikus számítógépének felhasználásával folytak. Az alapfeladat 92 változót és 57 feltételt tartalmazott, a magyar népgazdaság 1975. évi 15 szektoros bontású adatai alapján. Az eredeti feladatot 6 reflektorprogramozási feladattal helyettesítettük, amelyek mindegyike 2—3 ágazatot ölelt fel viszonylag részletesen. Valamennyi konvergenciakritérium alakulását figyelemmel kísértük, (5.2)-t is beleértve.

Az eredmények azt mutatták, hogy a reflektorprogramozás *konvergens*, mégpedig *az alábbi értelemben:*

1. Valamely programozási szakasz egymást követő iterációi monotonan konvergáltak.

2. A programozási szakaszok utolsó iterációi szintén monotonan konvergens sort alkottak.

3. Az eljárás nem véges, de viszonylag gyorsan (esetünkben 29 iterációval) a globális optimum igen jó (pár ezrelékes pontatlanságú) közelítését adta.

A reflektorprogramozás kedvező sajátossága az is, hogy már az első iterációtól kezdve közgazdaságilag jól értelmezhető eredményeket szolgáltat.

A reflektorprogramozás nagy előnye, hogy bármily nagyméretű feladatot képes szinte tetszőlegesen kis feladatok megoldásával helyettesíteni. Ez a körülmény valószínűleg hatékonyan felhasználható lesz mind a diszkrét, mind pedig a folytonos nemlineáris programozási feladatok, valamint más, például sztochasztikus programozási feladatok megoldása során.

(Beérkezett: 1972. szeptember 11.)

IRODALOM

1. Dantzig, G.—Wolfe, P.: Decomposition principle for linear programs. Operations Research, 1960. 8. 101—111. o.
2. Kornai J.—Lipták T.: Two-level planning. Econometrica, 1965. 33. 141—169. o.
3. Simon Gy.: Az árnyékárak viselkedésének tanulmányozása ex-post programozás alapján. MTA Közgazdaságtudományi Intézet, Budapest, 1967. 256—287. o.
4. Simon Gy.: Optimális tervezés reflektorprogramozással. A Gazdasági fejlődés és tervezés c. kötetben. Budapest, 1969. Közgazdasági és Jogi Könyvkiadó. 27—58. o.
5. Simon Gy.: Az optimális tervezés gazdasági mechanizmusának kérdései. MTA Közgazdaságtudományi Intézet. Budapest, 1970. 8—69. o.
6. Simon Gy.: Об экономическом механизме оптимального планирования. Periodica Polytechnica, Vol. 14. No. 4. 1970. 441—449. o.
7. Gábor Gy.: Beszámoló a reflektorprogramozás kísérleti számításairól. Tanulmány az OT Tervgazdasági Intézet részére. Budapest, 1971. (Kézirat.)
8. Krekó B.: Lineáris programozás. Közgazdasági és Jogi Könyvkiadó. Budapest, 1966.

THE PRINCIPLES AND ALGORITHM OF SEARCHLIGHT PROGRAMMING

Searchlight programming is a peculiar "decomposition" procedure for the approximative solution of large-scale linear programming problems. The essence of the procedure is that instead of the original large problem smaller ones are solved which are specially transformed from the large problem.

Searchlight programming can also be applied for linear programming problems where the constraints cannot be divided into central and sectoral groups. These is no need for a separate central model. The size of problems solvable with searchlight programming is in practice not limited.

The paper introduces the main principles of the procedure, first of all the "searchlight principle" playing a basic role, as well as the principle of combining primal and dual problems. It draws the scheme of searchlight programming models, the way of compiling them, the characteristics of information flow among the models. It deals with the convergence criteria of the iterative procedure, and refers to the experiences of test calculations made with searchlight programming.

ПРИНЦИПЫ И АЛГОРИТМ РЕФЛЕКТОР-ПРОГРАММИРОВАНИЯ

Рефлектор-программирование является своеобразным методом «разложения», разработанным автором, для приближающего решения больших задач линейного программирования. Суть метода, что вместо оригинальной большей задачи решаются малые задачи, которые являются специально трансформированными вариантами большей задачи.

Рефлектор-программирование можно примнять к задачам линейного программирования, в которых ограничивающие условия нельзя разделить на центральные и секторные условия. В особой центральной модели нет надобности. Величина задач, решаемых рефлектор-программированием на практике не ограничена.

Статья показывает главные принципы метода, прежде всего динамичный «рефлектор-принцип», сыграющий основную роль, а также принцип комбинирования примальных и дуальных задач. Намечает схему моделей рефлектор-программирования, способ их составлений, специальности обмена информацией между моделями. Он занимается критериями схождения итеративного ферлектор-программирования. Разрабатывается дальнейшее развитие процесса, которое сделает возможным его распространение на решение некоторых задач неличейного и дискретного программирования.