Forgó Ferenc

Dualitás és dekompozició egészértékű programozási feladatok esetében

Bevezetés

A lineáris programozás elmélete és a megoldó algoritmusok többsége valamilyen módon támaszkodik a feladat paramétereiből alkotott két programozási feladat, a primál és duál feladat közötti kapcsolatra. A duális változók árként („árnyékárként") való interpretációja és a közgazdasági elemzésekben való felhasználása már régóta vita tárgyát képezi. Anélkül, hogy ebben a vitában állást foglalnánk annyit feltétlenül megállapíthatunk, hogy a duális feladat, az árnyékárak hasznos információkat nyújtanak a probléma szerkezetéről és a közgazdasági elemzés hasznos eszközei.

A lineáris programozás dualitás elméletét többféle irányban is sikeresen általánosították [1] (konvex programozási feladatokra, végtelen dimenziós te rekre stb.). *Gomory* és *Baumol* [2] voltak az elsők, akik egészértékű lineáris programozási feladatok esetén is értelmeztek árnyékárakat és ki is számították őket. *Alcaly* és *Klevorick* [3] tökéletesítették a számítási módszert. Ezeknek a vizsgálatoknak a legfőbb korlátja, hogy a kapott árnyékárak függnek attól, hogy az eredeti egészértékű lineáris programozási feladatot milyen algoritmussal, sőt ezen belül is milyen lépéssorrenddel oldottuk meg. A legutóbbi időben *Bell* és *Shapiro* [13] egy konstruktív módszert ad meg, mely egy olyan duális feladatot generál, mely az eredeti egészértékű feladat optimális megoldása környezetében a lehetséges egész pontok konvex burkolóját közelíti meg. Algoritmusuk egyidejűleg oldja meg a primál és duál feladatot. *Balas* [4], [5] vegyes egészértékű feladatokkal foglalkozik. Az ő duális feladatai semmitmondóak a tiszta egészértékű esetben.

Ebben a cikkben egy, az eddigiektől különböző duális feladatot és árnyékárakat konstruálunk, melyek az alábbi legfontosabb tulajdonságokkal rendelkeznek:

1. Az árnyékárak közgazdaságilag interpretálhatók.
2. Megadható olyan algoritmus, mely a primál és duál feladatot egyidejűleg oldja meg.
3. A duális változók segítségével lehetővé válik speciális szerkezetű egészértékű programozási feladatok dekompozíciója. Két dekompozíciós módszert is adunk, melyek amellett, hogy számítási megtakarításokat tesznek lehetővé, az indirekt irányítás egy fajtájának modelljéül is szolgálhatnak.
4. Ez a dualítás igen egyszerűen terjeszthető ki bizonyos nem-lineáris egészértékű programozási feladatokra is.

A dolgozatban közölt algoritmusok semmiképpen sem tekinthetők kész, kiforrott és számítástechnikailag igazolt eljárásoknak. Az algoritmusok egy részének számítástechnikai kipróbálása jelenleg folyamatban van. A dolgozatban közölt eredmények inkább tekinthetők egy kutatási irány kiindulópontjának, mint kész, befejezett munkának.

1. Dualitás egészértékű programozási feladatok esetében

Az alábbi tiszta egészértékű lineáris programozási feladattal foglalkozunk:

$$\text{P:} \quad \begin{aligned} & z = cx \to \max \\ & Ax = b \\ & 0 \leq x \leq d, \quad A, b, d, c \text{ egészek} \\ & x = \text{egész,} \end{aligned} \tag{1}$$

ahol A egy m · n típusú mátrix, x,c,b,d pedig megfelelő dimenziójú vektorok˙ Az általánosság megsértése nélkül feltehetjük, hogy $b \geq 0$. Feltesszük azt is' hogy (1) lehetséges megoldásainak halmaza

$$L = \{x \mid A\, x = b,\ 0 \leq x \leq d,\ x = \text{egész}\}$$

nem üres. (Ez a feltétel a későbbiekben sok helyen feloldható, az egyszerű tárgyalhatóság miatt alkalmaztuk.) Mivel L korlátos, ezért (1) optimális megoldásainak halmaza, L_0 nem üres. Jelöljük z_0-al (1) optimális célfüggvény-értékét. Definiáljuk az L_u halmazt a következőképpen

$$L_u = \{x \mid uAx = ub,\ 0 \leq x \leq d,\ x = \text{egész}\},$$

ahol u egy nem-negatív egész vektor. (1) feladat feltételeit az u vektorral súlyozva egy feltételbe „sűrítettük" ezáltal. Nyilvánvaló, hogy ezért $L \subset L_u$. Legyen

$$U = \{u \mid u \geq 0,\ \max_{x \in L_u} cx = z_0\}$$

és

$$\overline{U} = \{u \mid u \geq 0,\ cx_0 = \max\{cx \mid x \in L_u\} \text{ és } x_0 \in L_u \Rightarrow x_0 \in L_0\}.$$

U mindazokat az $u \geq 0$ súlyokat tartalmazza, melyekkel egy feltételbe sűrítve (1) egyenlőségeit a cx célfüggvény-érték maximuma változatlan marad. $\overline{U}$ pedig azokat a nem-negatív súlyokat tartalmazza, melyekkel (1) feltételeit sűrítve az optimális megoldások halmaza változatlan marad. Világos, hogy $\overline{U} \subset U$, és ha valamely $u \in U$ esetén a

$$\begin{aligned} & cx \to \max \\ & x \in L_u \end{aligned} \tag{2}$$

feladatnak csak egy optimális megoldása van, akkor $\overline{U} = U$.

Az alábbi egészértékű programozási feladatot (1) duálisának fogjuk nevezni

$$D: \quad \begin{aligned} & ub \to \min \\ & u \in U \end{aligned} \tag{3}$$

a következő feladatot pedig (1) erős duálisának

$$\overline{D}: \quad \begin{aligned} & ub \to \min \\ & u \in \overline{U} \end{aligned}$$

A következő tétel $\overline{U}$ és ezáltal U nem ürességét biztosítja.

1. tétel. (*Bradley* [6]). Van olyan $\bar{u} > 0$, hogy $L_{\bar{u}} = L$.

Következmény. $\bar{u} \in \bar{U}$, mivel $\max_{x \in L_{\bar{u}}} cx = z_0$ és $L_{\bar{u}} = L$.

Mivel D és $\bar{D}$ célfüggvénye alulról korlátos (egy alsó korlát a 0), mind D-nek mind pedig $\bar{D}$-nek van optimális megoldása.

Fel szeretnénk hívni a figyelmet arra, hogy mind D-nek, mind pedig $\bar{D}$-nek a definíciója független attól, hogy milyen módszerrel oldottuk meg a primál P feladatot.

Minden $u \in \bar{U}$ vektort interpretálhatunk árrendszerként. Tegyük fel, hogy P egy gazdasági egység tevékenységét modellezi. Az $Ax = b$ feltételek a tevékenység ,,naturális'' korlátozottságát fejezik ki. Az u árrendszer segítségével a tevékenységet a (2) feladatban csak az $uAx = ub$,,pénzügyi'' feltétel, valamint az alsó és felső korlátok korlátozzák. Az u árrendszer biztosítja azt, hogy ha a pénzügyi feltételt kielégítjük, akkor a naturális feltételek automatikusan teljesülnek, amennyiben az x tevékenységvektor az optimális értéket veszi fel. Ezáltal a gazdasági egységet optimális működésre lehet ösztönözni egy árrendszerrel és azáltal, hogy a pénzügyi egyensúly betartását megköveteljük.

Ha $u \in U$, akkor csak azt állíthatjuk, hogy az x tevékenységvektornak van olyan optimális értéke, mely mellett a pénzügyi feltételek kielégítéséből a naturális feltételek kielégítése következik.

A fenti feltételeket kielégítő árrendszerekből természetesnek tűnik olyan $u \in \bar{U}$ (vagy $u \in U$) árakat választani, melyek az erőforrások összértékét, ub-t minimalizálják. Ez pedig pontosan az, amit D és $\bar{D}$ feladatban csináltunk.

A fenti interpretáció különösen jól illik olyan feladatokra, ahol P egy, a beruházási alternatívák közül válogató program. A beruházások területén pedig különösen hasznos egy olyan árrendszer, mely a pénzügyi és naturális feltételeket koordinálja.

A most bevezetett dualítás egy fontos tulajdonsága, hogy nem közvetlen általánosítása a lineáris programozás dualításelméletének. Ennek oka az hogy az 1. *tétel* nem igaz lineáris programozási feladatok esetén. Ennek ellenére, a lineáris programozás dualítás tételei közül egy, amely a marginális közgazdaságtanban alapvető, továbbra is érvényes marad.

Tegyük fel, hogy (1) i-ik feltétele az alábbi alakú

$$a_i'x + x_j = b_i, \qquad (a_i \geq 0) \tag{4}$$

ahol x_j egy egész kiegészítő változó, $c_j = 0$ és $a_{ij} = 0$. Ez azt jelenti, hogy (4) tulajdonképpen egyenlőtlenség:

$$a_i'x \leq b_i$$

Az i-ik erőforrást, melynek a korlátozottságát ez a feltétel kifejezi, szabad jószágnak nevezzük, ha (1) célfüggvény-értéke nem növelhető b_i növelésével.

2. tétel. Annak szükséges és elégséges feltétele, hogy az i-ik erőforrás szabad jószág legyen az, hogy létezik a D (vagy $\bar{D}$) feladatnak olyan $\bar{u}$ optimális megoldása, melynek az i-ik komponense $\bar{u}_i = 0$.

Bizonyítás. Tegyük először fel, hogy az i-ik jószág szabad, vagyis a

$$\begin{gathered} cx \to \max \\ A'x = b' \\ 0 \leq x \leq d \\ x = \text{egész} \end{gathered} \tag{5}$$

feladatnak ugyanaz az optimális célfüggvény-értéke, mint (1)-nek. Itt az $Ax = b$ feltételek közül az i-iket kihagytuk. Így ha $\bar{x}$ (1) optimális megoldása, akkor (5)-nek is optimális megoldása. (5) duálisának egy $\bar{u}$ megoldását az i-ik helyen egy 0-val kiegészítve D (vagy $\bar{D}$) egy olyan megoldását kapjuk, mely a kívánt tulajdonságú.

Ha viszont $\bar{u}$ a D-nek (vagy a $\bar{D}$-nek) egy megoldása és $\bar{u}_i = 0$, akkor (5)-nek ugyanaz az optimális célfüggvény-értéke, mint (1)-nek. Ez viszont éppen azt jelenti, hogy az i-ik jószág szabad.

Tegyük most fel, hogy $A \geq 0$. Legyen $\bar{u}$ D-nek (vagy a $\bar{D}$-nek) egy optimális megoldása. Egy tetszőleges $u \in U$ (vagy $u \in \bar{U}$) esetén a (2) feladat megoldása, mely tulajdonképpen egy hátizsák feladat:

$$\begin{gathered} cx \to \max \\ uAx = ub \\ 0 \leq x \leq d \\ x = \text{egész}, \end{gathered} \tag{6}$$

annál nehezebb, ha a leginkább használatos dinamikus programozást (lásd [7]) alkalmazzuk, minél nagyobb az ub jobboldal. Így a D és $\bar{D}$ feladatok megoldásaként kapott $\bar{u}$ a feltételek számítástechnikai szempontból vett legjobb súlyozása. *Bradley* [6] az (1) feladat megoldására a (6) hátizsák feladatra való visszavezetést javasolja olyan $\bar{u}$ szorzókkal, melyekre $L_{\bar{u}} = L$. Ezek a szorzók általában igen nagyok és jelenleg tisztázatlan még, hogy a feltételeknek ezekkel az $\bar{u}$ „*Bradley*-féle" szorzókkal egy feltételbe való sűrítése ad-e számítástechnikai előnyöket. Az optimális megoldás megkeresése szempontjából viszont felesleges, hogy $L_{\bar{u}} = L$ legyen. Ez az oka annak, hogy az általunk definiált duális változók nagyságrendekkel kisebbek lehetnek a *Bradley*-féle szorzóknál. Ezt látszik alátámasztani az a csekély számítástechnikai tapasztalat, ami eddig rendelkezésünkre áll. Természetesen az is lehet, hogy a duális változók még így is igen nagyok és a (6) hátizsák feladat megoldása nehezebb, mint az eredeti problémáé. Ezt a kérdést, úgy tűnik, hogy csak tapasztalati úton lehet tisztázni.

A következőkben felvázolunk egy algoritmust, mely egyidejűleg megoldja a P és a D (vagy $\bar{D}$) feladatot. Az algoritmus iterációkból áll. Minden egyes iterációban meghatározunk egy $u \geq 0$ vektort és megoldjuk a (6) hátizsák feladatot. Ha (6) optimális megoldása lehetséges megoldása (1)-nek, akkor egyúttal (1) optimális megoldása is.

Indulásképpen oldjuk meg az alábbi (triviális) feladatot:

$$\begin{gathered} ub \to \min \\ u \geq 0,\ u = \text{egész}. \end{gathered} \tag{7}$$

Legyen u_0 egy optimális megoldás. (Általában $u_0 = 0$.) Oldjuk meg az alábbi hátizsák feladatot:

$$cx \to \max$$

$$u_0 Ax = u_0 b \tag{8}$$

$$0 \leq x \leq d$$

$$x = \text{egész.}$$

Legyen x_0 (8) egy optimális megoldása. Ha $Ax_0 = b$, akkor x_0 nyilvánvalóan (1) egy optimális megoldása, u_0 pedig D optimális megoldása. Ha $Ax_0 \neq b$, akkor csatoljuk a következő feltételt (7) feltételrendszeréhez:

$$uAx_0 \neq ub \tag{9}$$

Ezáltal u_0-t kizártuk (7) lehetséges tartományából és ezért a következő lépésben (7) optimális megoldása, u_1 különböző lesz u_0-tól.

Általában a k-ik lépésben először megoldjuk az alábbi programozási feladatot:

$$ub \to \min$$

$$u \geq 0,\ u = \text{egész} \tag{10}$$

$$uAx_r \neq ub, \quad (r = 0, 1, \ldots, k-1).$$

Legyen u_k a (10)-nek egy optimális megoldása és oldjuk meg az alábbi hátizsák feladatot:

$$cx \to \max$$

$$u_k Ax = u_k b \tag{11}$$

$$0 \leq x \leq d,\ x = \text{egész.}$$

Ha (11) egy x_k optimális megoldása (1) lehetséges megoldása, akkor x_k P-nek, u_k pedig D-nek optimális megoldása. Ez azért igaz, mivel (10) lehetséges tartományából csak olyan u-kat zártunk ki, melyekre $u \notin U$. Mivel (11) lehetséges megoldásainak száma véges és $x_k \neq x_r$, $(r = 0,1, \ldots, k-1)$, az algoritmus véges számú lépésben konvergál.

Az algoritmus leírásánál nem tértünk ki azokra a problémákra, melyek a (10) típusú feladatok megoldása során keletkeznek elsősorban a szokatlan

$$uAx_r \neq ub \tag{12}$$

feltételek miatt. Mivel minden változó és konstans egész, ezért (12) azt jelenti, hogy az alábbi két egyenlőtlenség közül pontosan az egyiknek kell teljesülni:

$$uAx_r \leq ub - 1 \tag{13}$$

$$uAx_r \geq ub + 1 \tag{14}$$

Természetesnek tűnik ezáltal, hogy (10) feladatot a korlátozás és szétválasztás módszerével oldjuk meg. Mindannyiszor, amikor (10) feladathoz egy (12) típusú feltételt csatolunk, egy szétválasztást hajtunk végre (13) és (14)-nek megfelelően, a korlátozást és szétválasztást reprezentáló fa azon csúcspontjából, melyhez a minimális ub érték tartozik. A fa minden egyes csúcspontjához hozzárendelünk egy (13) vagy (14) típusú feltételt, egy nem-negatív, egész u vektort, mely az ub célfüggvényt minimalizálja a kérdéses csúcsponthoz és elődjeihez tartozó feltételek mellett, és egy B korlátot, mely ennek a programnak az optimális célfüggvény-értéke. (Ez definíciószerűen ∞, ha ennek a programnak nincs lehetséges megoldása.) Ezen kívül bizonyos csúcspontokhoz — azokhoz, amelyekből a szétválasztás megtörténik — a (11) típusú hátizsák feladat optimális megoldását is hozzárendeljük.

Az előzőekből nyilvánvaló, hogy ez a korlátozás és szétválasztás módszerén alapuló algoritmus is véges, feltéve, hogy az előforduló, egészértékű programozási feladatokat véges számú lépésben konvergáló algoritmusokkal oldjuk meg.

A módszer számítástechnikai realizációjával kapcsolatban két megjegyzést szeretnénk tenni.

1. Ha csupán a P primál feladat megoldására vagyunk kiváncsiak és az $u \in U$ vektorra csak azért van szükségünk, hogy P megoldását megkönnyítsük, akkor természetesen a számításokat kezdhetjük tetszőleges nem-negatív egész $\tilde{u}$-val. ($\tilde{u}$ lehet például D vagy $\bar{D}$ optimális megoldásának valamilyen becslése.) Ezt a becslést például (1) folytonos duál optimális megoldásából nyerhetjük alkalmas kerekítéssel vagy egyéb heurisztikus módszerrel. Ekkor viszont a (10) feladat feltételeihez az

$$ub \geq \tilde{u}b$$

egyenlőtlenséget is csatolnunk kell. Nyilván ahhoz, hogy (11) jól kezelhető legyen, $\tilde{u}b$-nek nem szabad nagynak lenni.

2. Ha csak a primál feladat megoldása érdekel bennünket, akkor megelégedhetünk a (10) feladat egy jó lehetséges megoldásával, ami általában sokkal egyszerűbben határozható meg, mint az optimális. Itt is egy jó lehetséges megoldáshoz kiindulópontul szolgálhatnak a korlátozás és szétválasztáshoz tartozó fa csúcspontjaihoz rendelt egészértékű programozási feladatok folytonos változatainak megoldásai (ezek lineáris programozási feladatok).

A primál és duál feladat egyidejű megoldására szolgáló fenti algoritmus illusztrációjaként oldjuk meg az alábbi feladatot:

$$\begin{aligned} z = 2x_1 + x_2 \quad - x_4 \quad + 3x_6 &\to \max \\ x_1 + 2x_2 + x_3 + x_4 + 2x_5 + 4x_6 &= 4 \\ 3x_1 \quad + 2x_3 + x_4 + x_5 + x_6 &= 3 \\ x_j \in \{0, 1\}, \ (j = 1, \ldots, 6) \end{aligned} \tag{15}$$

A megoldás menete legjobban a korlátozás és szétválasztáshoz tartozó fán követhető. Az u vektorokat meghatározó programok célfüggvénye mindig

$$4u_1 + 3u_2 \to \min.$$

A feltételeket, az optimális megoldásokat, a korlátokat, és ahol kiszámítottuk a (11) hátizsák feladat megoldásait mind feltüntettük a fa csúcspontjaiban.

Nincs feltétel
Optimális u megoldás: $\bar{u}_1 = 0$, $\bar{u}_2 = 0$. Korlát: $B = 0$
Optimális x megoldás: $\bar{x} = (1,1,0,0,0,1)$

$3u_1 + u_2 \leq -1$
Nincs lehetséges u, $B = \infty$

$3u_1 + u_2 \geq 1$
$\bar{u}_1 = 0$, $\bar{u}_2 = 1$, $B = 3$
$x = (0,1,1,0,0,1)$

$u_1 \leq -1$
Nincs lehetséges u, $B = \infty$

$u_1 \geq 1$
$\bar{u}_1 = 1$, $\bar{u}_2 = 0$, $B = 4$
$\bar{x} = (0,0,0,0,0,1)$

$u_2 \leq -1$
Nincs lehetséges u, $B = \infty$

$u_2 \geq 1$
$\bar{u}_1 = 1$, $\bar{u}_2 = 1$, $B = 7$
$\bar{x} = (0,1,0,0,0,1)$

$u_1 - u_2 \leq -1$
$\bar{u}_1 = 1$, $\bar{u}_2 = 2$, $B = 10$
$\bar{x} = (0,0,0,0,1,1)$

$u_1 - u_2 \geq 1$
$\bar{u}_1 = 2$, $\bar{u}_2 = 1$, $B = 11$
$\bar{x} = (0,1,1,1,0,0)$

Optimális megoldás

$-2u_1 + u_2 \geq 1$
$\bar{u}_1 = 1$, $\bar{u}_2 = 3$, $B = 13$

$2u_1 - u_2 \leq 1$
$\bar{u}_1 = 2$, $\bar{u}_2 = 3$, $B = 17$

Láthatjuk, hogy a primál optimális megoldás $x_0 = (0,1,1,1,0,0)$ az optimális célfüggvény-érték $z_0 = 0$, a duális feladat optimális megoldása $u_0 = (2,1)$ a hozzátartozó hátizsákprobléma pedig:

$$2x_1 + x_2 \qquad - x_4 \qquad + 3x_6 \rightarrow \max$$

$$5x_1 + 4x_2 + 4x_3 + 3x_4 + 5x_5 + 8x_6 = 11$$

$$x_j \in \{0,1\}, \quad (j = 1, \ldots, 6)$$

Az előzőekben tárgyalt tételek és módszerek könnyen általánosíthatók bizonyos nem-lineáris egészértékű feladatokra. Legyen P_N a következő nem-lineáris programozási feladat:

$$P_N: \qquad f(x) \to \max$$

$$g(x) = b \tag{16}$$

$$0 \leq x \leq d, \; x = \text{egész},$$

ahol $f(x)$ és $g(x)$ egészértékű minden egész x-re. Ekkor az L, L_u, U, $\bar{U}$ halmazokat és a D, $\bar{D}$ duál feladatokat a lineáris esettel analóg módon lehet definiálni. Az 1. tétel továbbra is érvényben marad (lásd *Bradley* [6]). Ugyanígy a 2. tétel is. A primál és duál feladatot egyidejűleg megoldó algoritmust is ugyanúgy lehet a (16) feladatra is alkalmazni. Természetesen, míg a (10) feladat

$$ub \to \min$$

$$u \geq O, \; u = \text{egész} \tag{17}$$

$$ug(x_r) \neq ub, \quad (r = 0, \ldots, k-1)$$

továbbra is lineáris marad, addig a (11) hátizsák feladat

$$f(x) \to \max$$

$$u_k\, g(x) = u_k b$$

$$0 \leq x \leq d, \; x = \text{egész}$$

nem-lineáris és általában csak akkor oldható meg hatékonyan, ha az f és g függvények szeparábilisak, vagyis

$$f(x) = \sum_{j=1}^{n} f_j(x_j)$$

$$g(x) = \sum_{j=1}^{n} g_j(x_j).$$

Ebben az esetben a dinamikus programozás módszere csaknem olyan hatékony, mint a lineáris esetben.

2. Egészértékű programozási feladatok dekompozíciója

Speciális struktúrájú lineáris programozási feladatok dekompozíciójával csaknem húsz éve foglalkoznak matematikusok, számítástechnikai szakemberek és közgazdászok. A dekompozíció megkönnyítheti a feladat megoldását, ugyanakkor modelljéül is szolgálhat egy központilag tervezett, de az alsóbb szinteken decentralizált gazdaságban végbemenő bizonyos döntési folyamatoknak is (lásd [8], [9], [11]). A problémát leggyakrabban az alábbi formában fogalmazzák meg:

$$L: \quad c_0 x_0 + c_1 x_1 + \ldots + c_r x_r \to \max$$

$$x_0, x_1, \ldots, x_r \geq 0$$

$$A_0 x_0 + A_1 x_1 + \ldots + A_r x_r = b_0 \tag{18}$$

$$\begin{aligned} B_1 x_1 \qquad &= b_1 \\ \ddots \qquad & \\ B_r x_r &= b_r. \end{aligned} \tag{19}$$

(18) az ún. ,,központi'' feltételeket, (19) pedig a ,,szektor'' feltételeket jelöli. A legtöbb dekompozíciós algoritmus egy ,,ár — termelés'' mechanizmussal dolgozik. Minden egyes iterációban a szektorok elkészítenek egy termelési tervet és ennek, valamint korábbi információknak a segítségével a központ meghatározza a központi erőforrások egy árrendszerét. A szektorok ezeken az árakon kapják a számukra szükséges központi erőforrásokat és módosítják termelési tervüket. (Részletesen lásd [8], [10], [11]-ben.) Ezt a mechanizmust lehet nem-lineáris konvex programozási feladatokra általánosítani [11], de nem-konvex és egészértékű problémák esetében eredeti formájában nem működik.

Az első részben bevezetett duális változókra támaszkodva két dekompozíciós algoritmust adunk meg speciális struktúrájú egészértékű programozási feladatok megoldására. Ezek természetesen alapvetően különböznek a lineáris programozásnál használt dekompozíciós eljárásoktól, de közgazdaságilag ugyancsak interpretálhatók és az eredeti feladat könnyebb megoldását elősegíthetik.

Természetesen itt is csak lehetőségről van szó, az algoritmusokban felhasznált duális változók nagyságrendje itt is kritikus szerepet játszik. Túl nagy duális változók azt jelenthetik, hogy az ismertetendő algoritmusok a gyakorlatban nem működnek.

Mostantól kezdve tegyük fel, hogy L valamennyi változója egész és korlátos, vagyis az

$$x_s = \text{egész}$$

$$0 \leq x_s \leq d_s, \quad (s = 0,1, \ldots, r)$$

feltételeket csatoljuk (18) és (19)-hez. Azt is feltesszük, hogy L-nek van lehetséges megoldása.

I. Algoritmus

Indulásképpen minden szektor megoldja a saját programját és annak duálisát:

$$c_s x_s \to \max$$

$$B_s x_s = b_s, \quad (s = 1, \ldots, r)$$

$$0 \leq x_s \leq d_s, \; x_s = \text{egész}.$$

Erre a célra lehet használni az 1. részben leírt algoritmust. Legyen egy primál és duál optimális megoldás $\bar{x}_s^0$ és u_s^0, $(s = 1, \ldots, r)$.

A szektorok az u_s^0 súlyokkal nyert ,,pénzügyi'' feltételüket elküldik a központnak, amely most az alábbi központi programot oldja meg:

$$\begin{aligned} c_0 x_0 + c_1 x_1 + \ldots + c_r x_r &\to \max \\ A_0 x_0 + A_1 x_1 + \ldots + A_r x_r &= b_0 \\ u_1^0 B_1 x_1 \qquad\qquad &= u_1^0 b_1 \\ \cdot\; \\ \cdot\; \\ u_r^0 B_r x_r &= u_r^0 b_r \\ O \leq x_s \leq d_s,\; x_s = \text{egész},\; &(s = 1, \ldots, r). \end{aligned} \tag{20}$$

A (20) feladatot bármilyen módszerrel megoldhatjuk, többek között az 1. rész módszerével is. Ha $x_0^0, x_1^0, \ldots, x_r^0$ a (20) optimális megoldása és mindegyik szektor-egyenletrendszert kielégíti, akkor L-nek is optimális megoldása. Ha x_s^0 nem lehetséges megoldása a s-ik szektorfeladatnak, akkor az

$$u_s B_s x_s^0 \neq u_s b_s \tag{21}$$

feltételt csatoljuk az s-ik szektor (10) típusú feladatához és új u_s^1 árnyékárakat határozunk meg. Ezek segítségével új pénzügyi feltételt küldünk a szektornak, amely új (20) típusú feladatot old meg stb.

Az általános lépés ugyanúgy megy. A szektorfeladatok duálisai (21) típusú feltételekkel bővülnek, és a központi program minden lépésben új megoldásokat szolgáltat. Ezért az algoritmus konvergens és véges számú lépésben meghatározza L egy optimális megoldását.

Érdemes megjegyezni, hogy ez az algoritmus bizonyos értelemben ,,fordítva'' működik mint a lineáris programozás legtöbb dekompozíciós algoritmusa. A szektorok árrendszert határoznak meg, pénzügyi feltételeiket a központnak küldik, amely termelési (beruházási) tervet határoz meg és ezt a megvalósíthatóság ellenőrzésére visszaküldi a szektornak. Ha a termelési (beruházási) terv valamelyik szektorban nem megvalósítható, akkor új pénzügyi feltételt határozunk meg és küldünk vissza a központnak, mindaddig, míg a központi program minden szektorban megvalósítható lesz.

Az algoritmust egy példával szemléltetjük. Oldjuk meg az alábbi programozási feladatot:

$$\begin{array}{rcl} 3x_1 - x_2 + 2x_3 + 4x_4 & - 2x_5 \qquad + 5x_7 - x_8 & \to \max \\ x_1 + 2x_2 + x_3 + 3x_4 & + x_5 + 2x_6 + 5x_7 + x_8 & = 10 \\ x_1 \qquad\quad + x_4 & \qquad\qquad + x_7 + x_8 & = 3 \\ \hline 2x_1 + x_2 + x_3 \qquad & & = 3 \\ x_3 + 2x_4 & & = 2 \\ \hline & x_5 + x_6 + 2x_7 + x_8 & = 3 \\ & x_6 + x_7 \qquad\quad & = 1 \end{array}$$

$$x_j \in \{0,1\}, \quad (j = 1, \ldots, 8)$$

Először oldjuk meg a két szektorprogramot

1.
$$\begin{aligned} 3x_1 - x_2 + 2x_3 + 4x_4 &\to \max \\ 2x_1 + x_2 + x_3 &= 3 \\ x_3 + 2x_4 &= 2 \\ x_j \in \{0,1\}, \quad (j = 1, \ldots, 4). \end{aligned} \tag{23}$$

Ennek optimális megoldása $\bar{x}_1^0 = (1,1,0,1)$, a duálisnak optimális megoldása pedig $u_1^0 = (1,1)$.

2.
$$\begin{aligned} -2x_5 + 5x_7 - x_8 &\to \max \\ x_5 + x_6 + 2x_7 + x_8 &= 3 \\ x_6 + x_7 &= 1 \\ x_j \in \{0,1\}, \quad (j = 5, \ldots, 8) \end{aligned} \tag{24}$$

A primál optimális megoldás $\bar{x}_2^0 = (0,0,1,1)$, a duál pedig $u_2^0 = (1,1)$.
Ezek után a központi program a következőképpen néz ki

$$\begin{aligned} 3x_1 - x_2 + 2x_3 + 4x_4 - 2x_5 + 5x_7 - x_8 &\to \max \\ x_1 + 2x_2 + x_3 + 3x_4 + x_5 + 2x_6 + 5x_7 + x_8 &= 10 \\ x_1 + x_4 + x_7 + x_8 &= 3 \\ 2x_1 + x_2 + 2x_3 + 2x_4 &= 5 \\ x_5 + 2x_6 + 3x_7 + x_8 &= 4 \\ x_j \in \{0,1\}, \quad (j = 1, \ldots, 8) \end{aligned} \tag{25}$$

Ennek optimális megoldása $x^0 = (1,1,1,0,0,0,1,1)$, ami az 1. szektor feltételeit nem elégíti ki. Ezért a

$$-u_1 + u_2 \neq 0$$

feltételt csatoljuk az 1. szektor duális feladatának feltétel rendszeréhez és új optimális megoldásként az $u_1^1 = (2,1)$ árnyékárakat kapjuk, melyek segítségével nyert

$$4x_1 + 2x_2 + 3x_3 + 2x_4 = 8$$

feltétellel helyettesítjük (25) harmadik egyenlőségét. Az ezek után nyert $x^1 = (1,1,0,1,1,1,0,1)$ optimális megoldás mindkét szektorban lehetséges és így a (22) feladat optimális megoldása.

II. Algoritmus

Ez az eljárás más elven nyugszik, mint az előző. Most először a központ határozza meg a központi erőforrások árnyékárait és egy paramétertől függő pénzügyi feltételt ír elő a szektoroknak. A szektorok egy parametrikus feladatot oldanak meg, melynek optimális célfüggvény-értékét (a paraméter függvényében) elküldik a központnak. A központ ezek figyelembevételével újra felosztja a rendelkezésre álló „pénzt'' (tulajdonképpen ez a paraméter) a szektorok között úgy, hogy az összhatékonyság növekedjék. Ha ez a program nem valósítható meg valamely központi naturális (eredeti) feltétel megsértése miatt, akkor a központi feladat duálisának egy új megoldását és ezáltal új pénzügyi feltételt határozunk meg és az egész eljárást megismételjük. Tesszük ezt mindaddig, amíg a szektorprogramok valamennyi központi feltételt is kielégítik.

Az egyszerűség kedvéért feltesszük, hogy $A_s \geq O$ $(s = 0,1, \ldots, r)$ és $b_0 \geq O$. Legelőször oldjuk meg a központi programot és annak duálisát, a szektorfeltételek figyelmen kívül hagyásával. Legyen egy optimális duál megoldás u_0^0, az ezzel nyert pénzügyi feltétel pedig

$$u_0^0 A_0 x_0 + u_0^0 A_1 x_1 + \ldots + u_0^0 A_r x_r = u_0^0 b_0.$$

Jelöljük ezt az egyenlőséget egyszerűen a következőképpen

$$p_0^0 x_0 + p_1^0 x_1 + \ldots + p_r^0 x_r = t^0 \tag{26}$$

A következő lépésben megoldjuk az alábbi parametrikus szektorprogramokat

$$\begin{gathered} c_s x_s \to \max \\ B_s x_s = b_s \\ p_s^0 x_s = z_s, \qquad (s = 1, \ldots, r) \\ 0 \leq x_s \leq d_s,\ x_s = \text{egész}, \end{gathered} \tag{27}$$

és az alábbi feladatot

$$\begin{gathered} c_0 x_0 \to \max \\ p_0^0 x_0 = z_0 \\ 0 \leq x_0 \leq d_0,\ x_0 = \text{egész}, \end{gathered} \tag{28}$$

ahol $z_0, z_1, \ldots, z_r$ egész paraméterek. A (27) és (28) feladatok megoldására a dinamikus programozás módszerét lehet használni kombinálva a nem-parametrikus feltételek duális változók segítségével való egy feltételbe sűrítésével. (27) és (28) megoldásaként az

$$\begin{gathered} f_s(z_s), \quad (s = 0,1, \ldots, r) \\ 0 \leq z_s \leq t^0,\ z_s = \text{egész} \end{gathered}$$

optimális célfüggvény-érték függvényeket kapjuk. (Definíciószerűen $f_s(z_s) = -\infty$, ha a szektorprogram nem megoldható valamely z_s-re).

Itt jegyezzük meg, hogy ha az $A_s \geq O$ feltételt elhagyjuk, akkor z_s alsó és felső korlátai változnak csak, egyebekben az egész eljárás változatlan marad.

Oldjuk meg most a következő központi erőforrás elosztó programot:

$$\begin{gathered} F(z) = f_0(z_0) + f_1(z_1) + \ldots + f_r(z_r) \to \max \\ z_0 + z_1 + \ldots + z_r = t^0 \\ z_0, z_1, \ldots z_r \geq 0,\ z_0, z_1, \ldots z_r = \text{egész} \end{gathered} \tag{29}$$

Ezt a feladatot a dinamikus programozás módszerével oldhatjuk meg. Legyen $z_0^0, z_1^0, \ldots z_r^0$ egy optimális megoldás és $x_0^0, x_1^0, \ldots x_r^0$ a hozzátartozó megoldásvektorok, melyek nyilván a szektorok lehetséges megoldásai, $F(z^0) = -\infty$ nem állhat fenn, hiszen L-nek van legalább egy lehetséges megoldása. Ha $x_0^0, x_1^0, \ldots, x_r^0$ kielégíti a központi feltételeket, akkor az L egy optimális megoldása. Ha nem, akkor egy

$$u_0 A_0 x_0^0 + u_0 A_1 x_0^1 + \ldots + u_0 A_r x_r^0 \neq u_0 b \tag{30}$$

feltételt csatolunk a központi feladat duálisának feltételrendszeréhez és új u_0^1 árnyékárakat határozunk meg, melyekkel egy új központi pénzügyi feltételt nyerhetünk és az egész eljárást megismételhetjük.

Mivel minden lépésben L legalább egy lehetséges megoldását kizárjuk, ezért ez az algoritmus is véges számú lépésben konvergál.

Oldjuk meg most a (22) feladatot ezzel a módszerrel is. Indulásképpen az $u_1^0 = 0$ és $u_2^0 = 0$ árnyékárakkal képezzük a pénzügyi feltételeket és utána megoldjuk a szektorprogramokat:

1.
$$\begin{aligned} 3x_1 - x_2 + 2x_3 + 4x_4 &\to \max \\ 0x_1 + 0x_2 + 0x_3 + 0x_4 &= z_1 \\ 2x_1 + x_2 + x_3 &= 3 \\ x_3 + 2x_4 &= 2 \\ x_j \in \{0,1\}, \quad (j = 1, \ldots, 4); \end{aligned}$$

2.
$$\begin{aligned} -2x_5 + 5x_7 - x_8 &\to \max \\ 0x_5 + 0x_6 + 0x_7 + 0x_8 &= z_2 \\ x_5 + x_6 + 2x_7 + x_8 &= 3 \\ x_6 + x_7 &= 1 \\ x_j \in \{0,1\}, \quad (j = 5, \ldots, 8). \end{aligned}$$

Mivel z_1 és z_2 csak a 0 értéket veheti fel, $f_1(0) = 6$ és $f_2(0) = 4$, a megfelelő optimális megoldások pedig

$$x_1^0 = (1,1,0,1)$$
$$x_2^0 = (0,0,1,1),$$

amelyek nem elégítik ki a központi feltételeket és így egy új

$$2u_1 - u_2 \neq 0$$

kikötést teszünk a duál változókra. Az új duál változók $u_1^1 = 0$, $u_2^1 = 1$, a szektor programok pedig az alábbiak:

1.
$$\begin{aligned} 3x_1 - x_2 + 2x_3 + 4x_4 &\to \max \\ x_2 + x_4 &= z_1 \\ 2x_1 + x_2 + x_3 &= 3 \\ x_3 + 2x_4 &= 2 \\ x_j \in \{0,1\}, \quad (j = 1, \ldots, 4); \end{aligned}$$

$$\text{2.} \qquad \begin{aligned} -2x_5 \qquad + 5x_7 - \quad x_8 &\to \max \\ x_7 + \quad x_8 &= z_2 \\ x_5 + x_6 + 2x_7 + \quad x_8 &= 3 \\ x_6 + x_7 \qquad &= 1 \\ x_j \in \{0,1\}, \quad (j = 5, \ldots, 8). \end{aligned}$$

Az $f_1(z_1)$ és $f_2(z_2)$ függvények az alábbiak

	$f_1(z_1)$
0	$-\infty$
1	$-\infty$
2	6
3	$-\infty$

z_2	$f_2(z_2)$
0	$-\infty$
1	3
2	4
3	$-\infty$

A központi erőforráselosztó program a következő

$$f_1(z_1) + f_2(z_2) \to \max$$
$$z_1 + z_2 = 3$$
$$z_1, z_2 \geq 0, \quad z_1, z_2 = \text{egész},$$

melynek az optimális megoldása $z_1 = 2$, $z_2 = 1$; a megfelelő megoldásvektorok pedig

$$x_1^1 = (1,1,0,1)$$
$$x_2^1 = (1,0,1,0),$$

melyek szintén nem elégítik ki a szektor feltételeket. Ezen megoldások segítségével egy

$$u_1 \neq 0$$

feltételt csatolhatunk a központi feladat duáljához s így az $u_1^2 = 1$, $u_2^2 = 0$ új árnyékárakhoz jutunk, melyek az alábbi szektorprogramokhoz vezetnek:

$$\text{1.} \qquad \begin{aligned} 3x_1 - \quad x_2 + 2x_3 + 4x_4 &\to \max \\ x_1 + 2x_2 + \quad x_3 + 3x_4 &= z_1 \\ 2x_1 + \quad x_2 + \quad x_3 \qquad &= 3 \\ x_3 + 2x_4 &= 2 \\ x_j \in \{0,1\}, \quad (j = 1, \ldots, 4); \end{aligned}$$

$$\text{2.} \qquad \begin{aligned} -2x_5 \qquad + 5x_7 - \quad x_8 &\to \max \\ x_5 + 2x_6 + 5x_7 + \quad x_8 &= z_2 \\ x_5 + x_6 + 2x_7 + \quad x_8 &= 3 \\ x_6 + x_7 \qquad &= 1 \\ x_j \in \{0,1\}, \quad (j = 5, \ldots, 8). \end{aligned}$$

Az $f_1(z_1)$ és $f_2(z_2)$ függvények az alábbiak

z_1	$f_1(z_1)$
0	$-\infty$
1	$-\infty$
2	$-\infty$
3	$-\infty$
4	$-\infty$
5	$-\infty$
6	6
7	$-\infty$
8	∞
9	∞
10	∞

z_2	(z_2)
0	$-\infty$
1	$-\infty$
2	$-\infty$
3	$-\infty$
4	-3
5	$-\infty$
6	**4**
7	$-\infty$
8	$-\infty$
9	$-\infty$
10	$-\infty$

Az

$$f_1(z_1) + f_2(z_2)$$

$$z_1 + z_2 = 10$$

$$z_1, z_2 \geq 0, \quad z_1, z_2 = \text{egész}$$

erőforráselosztó program optimális megoldása $z_1 = 6$, $z_2 = 4$, melyekhez az

$$x_1^2 = (1,1,0,1)$$

$$x_2^2 = (1,1,0,1)$$

megoldásvektorok tartoznak. Ezek kielégítik a központi feltételeket és így L feladat optimális megoldását adják.

Végezetül meg kell említenünk, hogy mindkét dekompozíciós algoritmust könnyen ki lehet terjeszteni olyan nem-lineáris egészértékű feladatokra, melyekkel az 1. részben is foglalkoztunk.

(Beérkezett: 1976. február 6.)

IRODALOM

1. MANGASARIAN, O. L.: *Nonlinear programming.* New York, 1969. McGraw-Hill.
2. GOMORY, E. R.—BAUMOL, W. J.: *Integer programming and pricing.* Econometrica 28, 1960. pp. 521—550.
3. ALCALY, R. F.—KLEVORICK, A. K.: *A note on the dual prices of integer programs.* Econometrica 34, 1966. pp. 206—214.
4. BALAS, E.: *Duality in discrete programming.* Stanford University, Operations Research House, Technical Report, No. 67—5. 1967.
5. BALAS, E.: *A duality theorem and an algorithm for (mixed) integer nonlinear programming.* Linear Algebra and Application 4, 1971. pp. 341—352.
6. BRADLEY, G. H.: *Transformation of integer programs to knapsack problems.* Discrete Mathematics 1, 1971. No. 1. pp. 29—45.
7. GARFINKEL, R. S.—NEMHAUSER, G. L.: *Integer programming.* New York, 1972. John Wiley and Sons.
8. DANTZIG, G. B.—WOLFE, P.: *Decomposition principle for linear programs.* Operations Research 8, 1960. pp. 101—111.
9. KÜNZI, H. P.—TAN, S.: *Lineare Optimierung grosser Systeme.* Berlin—Heidelberg—New York, 1966. Springer Verlag.
10. DANTZIG, G. B.: *Linear programming and extensions.* Princeton, N. J. 1963. Princeton University Press.

11. LASDON, L. S.: *Optimization theory for large systems.* New York, 1972. The Macmillan Company.
12. SCHRAGE, L.: *Using decomposition in integer programming.* Naval Research Logistics Quarterly, Vol. 20. No. 3. (1973) pp. 469–476.
13. BELL, E. D.–SHAPIRO, J. F.: *A finitely convergent duality theory for zero-one integer programming.* MIT Operations Research Center, OR 043–75, May 1975.
14. FORGÓ, F.: *Shadow prices and decomposition for integer programs.* Budapest, 1975. Dept. of Mathematics, Karl Marx University of Economics.

DUALITY AND DECOMPOSITION IN INTEGER PROGRAMMING

To tne integer linear programming (primal program) problem we define a dual program. The optimum solutions (shadow prices) of the dual program are economically interpreted, and an algorithm is given to solve the primal and dual programs simultaneously. The introduction of the dual variables enables the decomposition of some specially-structured integer programming problems. Two decomposition methods are presented which, in addition to potentially yielding computational advantages, may even serve as a model for a type of indirect control. This kind of duality can very simply be extended to certain types of non-linear integer programming problems.

The results offered in the study are to be considered (primarily from computational aspects) as a starting-point for a research line rather than as a finished work.

ДВОЙСТВЕННОСТЬ И ДЕКОМПОЗИЦИЯ В ЗАДАЧАХ ЦЕЛОЧИСЛЕННОГО ПРОГРАММИРОВАНИЯ

К задаче целочисленного линейного программирования (прямая задача) определим двойственную задачу. Дадим экономическую интерпретацию оптимальных решений (теневых цен) двойственной задачи, далее определим алгоритм, который совместно решает и прямую и двойственную задачи. С помощью двойственных переменных станет возможным декомпозиция целочисленных задач программирования со специальной структурой. В статье мы показываем два декомпозиционных метода, которые, кроме того, что могут иметь некоторые потенциальные счетно-технические преимущества, могут служить и моделью определенного вида косвенного управления. Такой вид двойственности легко распространяется на некоторые задачи нелинейного целочисленного программирования.

Результаты, излагаемые в данной статье, скорее можно назвать (в первую очередь с точки зрения счетной техники) исходным пунктом направления некоторых исследований, чем готовой, завершенной работой.