Mészáros Katalin

A PARTICIONÁLÁS ALKALMAZÁSÁNAK JELENTŐSÉGE, KÜLÖNÖS TEKINTETTEL A VEGYES EGÉSZÉRTÉKŰ FELADATOKRA

BEVEZETŐ

A matematikai programozási feladatok körében fontos szempont a kidolgozott modellek megoldhatósága a reális lehetőségek határain belül. A lineáris modellek és módszerek a nemlineárisokkal szemben két szempontból előnyösebbek:

a) közgazdaságilag jól interpretálhatók,

b) matematikailag viszonylag könnyen kezelhetők.

Ezek az előnyök azonban nem ellensúlyozzák azt a tényt, hogy a lineáris modellekkel a valóságos viszonyokat jól megközelíteni csak többé-kevésbé tudjuk. Folytonos programozási feladatok esetében a matematikai, számítástechnikai nehézségek ugrásszerűen nőnek, ha feltételek és a célfüggvény vonatkozásában a linearitást gyengébb feltétellel helyettesítjük. Hasonló nehézségekkel kell számolnunk az integer programozási feladatoknál is, azzal a különbséggel, hogy itt már a lineáris típusú feladat megoldása is problematikus lehet. Ebből kifolyólag semmi esetre sem hagyható figyelmen kívül az egyes matematikai módszerek gyakorlati alkalmazása és számszerűsíthetősége. Ebben nagy segítséget nyújt a particionáló eljárás, amely minden olyan esetben alkalmazható, amikor egy matematikai programozási feladat feltételeiben egy lineáris és egy nemlineáris rész szerepel, melyeknek nincs közös változójuk, valamint a nemlineáris résznek megfelelő feladat külön is megoldható. A matematikai programozási feladat ez esetben szétválasztható egy közönséges lineáris programozási feladatra, amely a jól kezelhető és a viszonylag könnyen megoldható problémák körébe tartozik, és egy matematikai programozási feladatra, amellyel már sokkal alacsonyabb dimenzióban megbirkózhatunk, mint az eredeti feladattal. Már ennyi érv is elegendőnek tűnik ahhoz, hogy indokolt legyen a bonyolult matematikai programozási feladatok szétbontása két alproblémára a könnyebb megoldhatóság érdekében. Az ára pedig nem más, mint az alproblémák egymásutáni felváltott megoldása a paraméterek bizonyos értékei mellett. A particionáló eljárás alkalmazásakor a feladatban szereplő változók eleve két részre bonthatók, az ún. lineáris változókra — ezen elnevezésen a lineáris típusú folytonos változókat értjük — és a többi változókra, amely lehet:

a) lineáris típusú diszkrét értékű változó,

b) nemlineáris típusú folytonos változó,

c) nemlineáris típusú diszkrét értékű változó.

A particionáló eljárás tehát a következő esetekben alkalmazható:

a) lineáris típusú vegyes egészértékű feladatoknál,

b) csak részben lineáris típusú folytonos változós feladatoknál,

c) lineáris típusú folytonos változós és nemlineáris típusú diszkrét változós feladatoknál, amennyiben a matematikai programozási alprobléma a már megoldható problémák körébe tartozik. Természetesen a lineáris programozási alprobléma megoldhatóságához nem fér kétség. A particionáló eljárás alkalmazható még a tiszta lineáris programozási feladatokra is, ekkor azonban a Dantzig—Wolfe-féle dekompozíciós eljáráshoz hasonló módszert kapunk. (Megállapítható, hogy bizonyos rokonság van a particionáló eljárás és a lineáris modellekkel kapcsolatos dekompozíciós módszer között.) Lényeges különbség a particionáló eljárás és a Dantzig—Wolfe-féle dekompozíciós módszer között, hogy a particionáláskor a változókat osztjuk két csoportra, míg a Dantzig—Wolfe dekompozíció esetén a feltételeket. A particionáló módszer alkalmazásának közönséges lineáris programozási feladatra általában nincs előnye, mert a módszer akkor gazdaságos, ha a matematikai programozási feladatban a lineáris típusú folytonos változón kívül másfajta változó is szerepel.

A matematikai programozási feladatok változók szerinti felbontásának problémájával Benders foglalkozott. A megoldás menetének felkutatása is az ő nevéhez fűződik.

A vegyes egészértékű feladatok felbontása.

A vegyes egészértékű programozási feladat néven ismert matematikai problémának a következő általános matematikai modell felel meg:

$$\max \quad \{c_1^* x + c_2^* y \mid x \in E_{n_1}^+,\ y \in I_{n_2}^+,\ A_1 x + A_2 y \leq b\} \tag{1}$$

vagy

$$\min \quad \{c_1^* x + c_2^* y \mid x \in E_{n_1}^+,\ y \in I_{n_2}^+,\ A_1 x + A_2 y \geq b\} \tag{2}$$

A kifejezésekben az $E_{n_1}^+$ az n_1 dimenziós euklideszi tér azon vektorainak halmazát jelenti, amelynek koordinátái nem negatív számok. Az $I_{n_2}^+$ az n_2 dimenziós euklideszi tér azon vektorainak halmazát jelenti, amelyek koordinátái nem negatív egész számok.

Amennyiben figyelembe vesszük, hogy (1) bármikor (2)-vé alakítható, és fordítva, (2) (1)-gyé alakítható, valamint, hogy bármely egészértékű változó visszavezethető 0–1 értékű változókra, akkor további vizsgálataink tárgyát a következő feladattípus képezi:

$$\max \quad \{c_1^* x + c_2^* y \mid x \in E_{n_1}^+,\ y \in B_{n_2},\ A_1 x + A_2 y \leq b\}, \tag{3}$$

ahol a B_{n_2} szimbólummal az n_2 dimenziós euklideszi tér azon vektorainak halmazát jelöltük, amelyek komponensei O vagy 1 értékűek.

A Benders által kidolgozott particionálási eljárás alkalmazható akkor is, ha történetesen minden változó lineáris típusú, mint a mi esetünkben is, amikor a vegyes egészértékű feladat felbontása a cél, a könnyebb megoldhatóság érdekében. Ez az eljárás megköveteli két alaprobléma szukcesszív megoldását. A (3) feladat felbontódik egy közönséges lineáris programozási problémára és egy lineáris típusú vegyes egészértékű problémára n_2 darab 0–1 értékű változóval és egy tetszőleges (nem feltétlenül pozitív) változóval. Bevezetve az y_0 skalárt a

$$c_1^* x + c_2^* y \to \max$$

célfüggvény a következő alakot ölti

$$y_0 \to \max$$

$$c_1^* x + c_2^* y \geq y_0 .$$

A (3) probléma a kovetkezőképpen osztható fel az előbbiekben említett két alaproblémára: P_1-re és P_2-re.

P_1

$$y_0 \to \max$$

$$y_0 - c_2^* y + u_r^* A_2 y \leq u_r^* b \qquad (r = 1, 2, \ldots, t)$$

$$u_s^* A_2 y \leq u_s^* b \qquad (s = 1, 2, \ldots, w)$$

$$y \in B_{n_2}$$
$$y_0 = \text{tetszőleges}$$

P_2

$$c_1^* x \to \max$$

$$A_1 x \leq b - A_2 \bar{y}$$

$$x \geq 0$$

ahol $c_1=(c_1^j)$; $c_2=(c_2^k)$; $A_1=(a_1^{ij})$; $A_2=(a_2^{ik})$, $b=(b_i)$ $j=1, 2, \ldots, n_1$; $k=1, 2, \ldots, n_2$ és $i=1, 2, \ldots, m$.

Az u_r vektort az $\{u|u^*A_1 \geqq c_1,\ u \geqq O\}$ halmaz extremális pontjai alkotják, azaz P_2 duálisának bázismegoldásai. Az u_s vektort pedig az $\{u|u^*A_1 \geqq O,\ u \geqq O\}$ kónusz extremális irányai alkotják. Az $\tilde{y}$ vektor a P_2 feladatban a P_1 feladat optimális megoldáspárjának, $(\tilde{y}_0, \tilde{y})$-nak az egyik komponense.

A (3) probléma megoldásának két fázisa van; P_1 és P_2 problémák megoldása. Egymásra következő megoldogatásuk elvezet a (3) probléma megoldásához. A P_1 megoldása $(\tilde{y}_0, \tilde{y})$, $\tilde{y}$ a P_2-be helyettesítendő, P_2 ezután megoldandó. A P_2 megoldásától függően a P_1 egy vagy két sorral bővül. Most ismét P_1 megoldása van soron, majd P_2. Pontokba foglalva a particionálás egyes lépéseit az algoritmus így adható meg:

A fázis. Megoldjuk a P_1-et.

A1. Ha a P_1-nek nincs lehetséges megoldása, akkor a (3)-nak sincs.

A2. Ha a P_1-nek van optimális megoldása $(\tilde{y}_0, \tilde{y})$, vagy nincsen véges optimuma, de $(\tilde{y}_0, \tilde{y})$ lehetséges megoldás bármilyen nagy értéket is vesz fel $\tilde{y}_0$ és új jobb oldalt definiál a P_2-ben. A B fázis következik.

B fázis. Megoldjuk a P_2-t.

B1. Ha a P_2-nek nincs véges optimuma, akkor a (3)-nak sincs.

B2. Ha a P_2 optimális megoldása $\tilde{x}$ olyan, hogy $c_1^*\tilde{x} \geqq \tilde{y}_0 - c_2^*\tilde{y}$, akkor $(\tilde{x}, \tilde{y})$ a (3) optimális megoldását képezi. Ha $c_1^*\tilde{x} < \tilde{y}_0 - c_2^*\tilde{y}$, akkor u_r a P_2 duálisának optimális megoldása új feltétellel bővíti a P_1-et. Így az A fázis következik.

B3. Ha a P_2-nek nincs lehetséges megoldása, akkor u_r (ahol u_r a P_2 duálisának megoldása, azaz az $\{u|u^*A_1 \geqq c_1,\ u \geqq 0\}$ halmaz extremális pontja) segítségével egy feltétellel bővítjük a P_1-et, u_s segítségével pedig még egy feltétellel bővítjük P_1-et (ahol u_s az $\{u|u^*A_1 \geqq O,\ u \geqq 0\}$ kónusz extremális iránya). Ezután visszatérünk az A fázisra.

Az ismertetett particionálási eljárás folyamatábrán is szemléltethető. (1. ábra)

A P_1 problémával kapcsolatban érdemes felfigyelni a következő sajátosságokra:

a) a 0—1 értékű y_k ($k= 1, 2, \ldots, n_2$) változók mellett y_0 nem korlátos változót is tartalmaz,

b) y_0 együtthatója pozitív vagy nulla minden $\leqq$ feltételben,

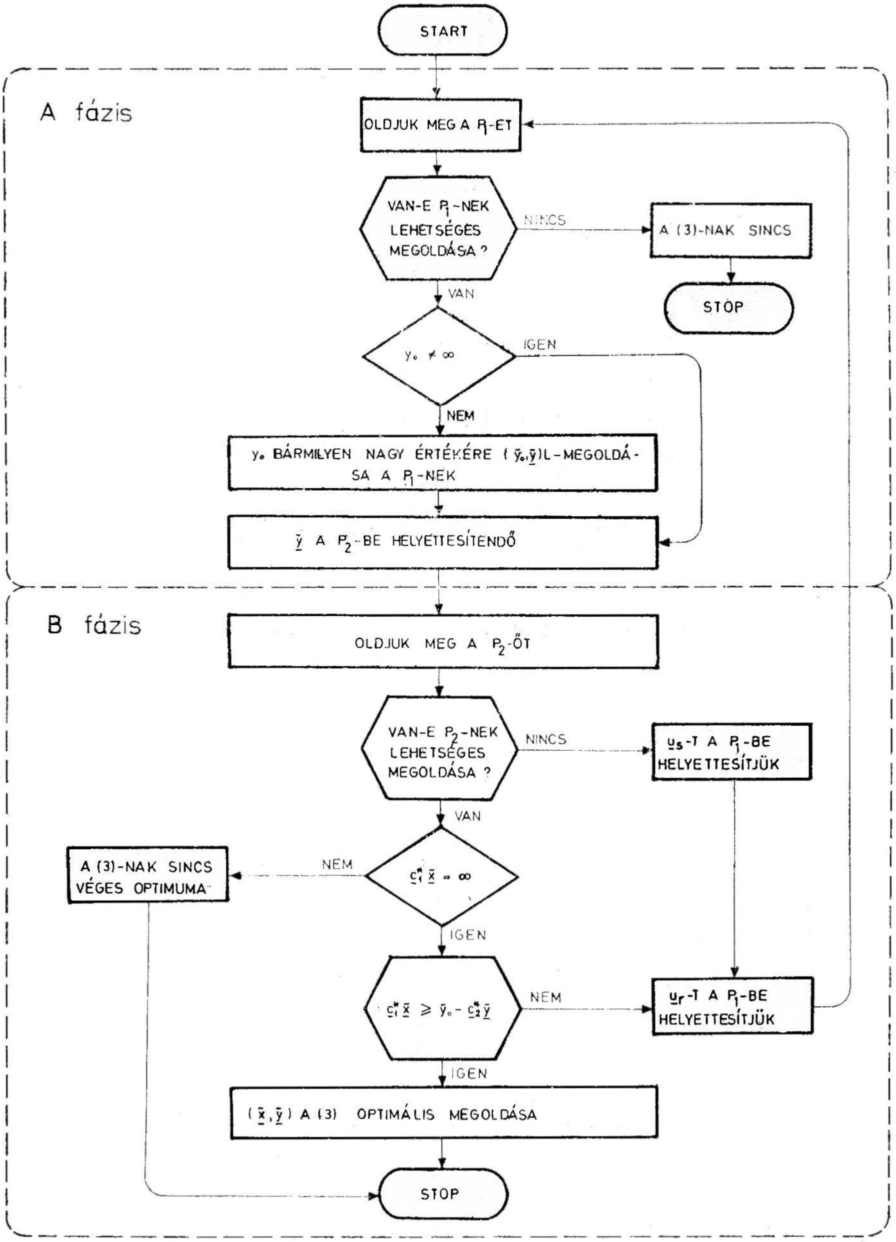
c) a célfüggvény nem tartalmaz egyetlenegy 0—1 értékű y_k változót sem,

d) ismételten megoldásra kerül ugyanaz a probléma egy vagy két feltétel hozzáadásával.

A P_2 probléma közönséges lineáris programozási probléma, amely a szimplex módszerrel minden nehézség nélkül megoldható.

A vegyes egészértékű feladatok megoldására számos módszer létezik. A feladatok jellegétől függően adott esetben a metszési és kombinatorikus

1. ÁBRA



START
A fázis
OLDJUK MEG A P1-ET
VAN-E P1-NEK LEHETSÉGES MEGOLDÁSA ?
NINCS
A (3)-NAK SINCS
STOP
VAN
$y_0 \neq \infty$
IGEN
NEM
y_0 BÁRMILYEN NAGY ÉRTÉKÉRE $(\tilde{y}_0, \tilde{\underline{y}})$L-MEGOLDÁSA A P1-NEK
$\tilde{\underline{y}}$ A P2-BE HELYETTESÍTENDŐ
B fázis
OLDJUK MEG A P2-ŐT
VAN-E P2-NEK LEHETSÉGES MEGOLDÁSA ?
NINCS
$\underline{u}_s$-T A P1-BE HELYETTESÍTJÜK
VAN
A (3)-NAK SINCS VÉGES OPTIMUMA
NEM
$\underline{c}_1^* \tilde{\underline{x}} = \infty$
IGEN
$\underline{c}_1^* \tilde{\underline{x}} \geq \tilde{y}_0 - \underline{c}_2^* \tilde{\underline{y}}$
NEM
$\underline{u}_r$-T A P1-BE HELYETTESÍTJÜK
IGEN
$(\tilde{\underline{x}}, \tilde{\underline{y}})$ A (3) OPTIMÁLIS MEGOLDÁSA
STOP

módszerek közül kell kiválasztani a megfelelőt. Mindezek a módszerek működnek a particionálás végrehajtása nélkül is. Egyedüli kivétel a módosított filter módszer, amely a feladat Benders szerinti particionálását is magába foglalja. Mégis célszerű a megismert particionálási eljárásra támaszkodva megoldani a vegyes egészértékű feladatokat. Előnyei a következők:

a) nagyobb méretű feladatoknál remény van rá, hogy ily módon hamarabb jutunk eredményhez — kevesebb az elhasznált gépidő és kisebb a memóriaigény,

d) amikor a nemfolytonos, tehát P_1 feladat megoldásáról van szó. csupán egyetlen — jelen esetben y_0-val jelölt — folytonos változó fordul elő (P_1 probléma a tulajdonsága szerint). Habár alkalmazni kell a vegyes egészértékű feladatok megoldására szolgáló algoritmust, ez mégis egyszerűbb az eredeti feladat megoldásánál,

c) a P_1 probléma d) tulajdonsága már önmagában véve is előnyt jelent,

d) nem győzzük eleget hangsúlyozni, hogy milyen nagy könnyítés a szakember számára a P_2 probléma megoldhatósága a szimplex módszerrel, amely általában megtalálható a számítógépek programkönyvtárában.

A számítástechnikai hatékonyság természetesen attól függ, hogy milyen gyorsan lehet a P_1 és P_2 feladatokat megoldani, és mindezt hányszor kell megismételni. Elsősorban P_1 probléma mérete a korlátozó tényező. Néhány nagyobb feladat tapasztalatai kedvezőek. Közgazdasági alkalmazásoknál (például termelési-beruházási feladatoknál) az egészértékű változók száma viszonylag alacsony a folytonos változókéhoz képest. Az ilyen típusú feladatok esetében nagy előnnyel jár a particionáló eljárás alkalmazása.

ÖSSZEFOGLALÁS

Amikor a matematikai programozási feladatokat megoldandó algoritmusok jóságáról esik szó, akkor elsősorban azoknak számítógépre programozhatóságát vizsgálják. Egy algoritmus ugyanis annál jobb, minél kisebb memóriaigénnyel és minél rövidebb idő alatt old meg egy adott típusú feladatot. A Benders-féle particionálás használatakor is ez a cél. Konkrét számítástechnikai tapasztalatok főleg a vegyes egészértékű feladatokkal vannak. Ezek azt mutatják, hogy nő az algoritmusok hatékonysága, ha a feladatokat előzőleg particionáljuk.

További kutatások tárgyát képezi a többi matematikai programozási algoritmus hatékonyságának megállapítása számítástechnikai vizsgálattal alkalmazva a Benders szerinti felbontást.

Felhasznált irodalom

1. Krekó Béla: Optimumszámítás. Közgazdasági és Jogi Könyvkiadó, Budapest, 1972.
2. Kovács Béla: A diszkrét programozás kombinatorikus módszerei. Bolyai J. Matematikai Társulat, Budapest, 1969.
3. Forgó Ferenc: Nemkonvex és diszkrét programozás. Közgazdasági és Jogi Könyvkiadó, Budapest, 1978.
4. Mészáros Katalin: A filter eljárásra épülő algoritmus számítástechnikai vizsgálata. Magiszteri dolgozat, Szabadka, 1982.

Rezime

Značaj primene particioniranja sa posebnim osvrtom na celobrojne — mešovite probleme

Kada je reč o efikasnosti algoritma za rešavanje problema matematičkog programiranja, onda se prvenstveno misli na to, da li su algoritmi pogodni za programiranje na računaru. Efikasnost algoritama se određuje na osnovu potrošenog računarskog vremena i na osnovu zahteva za memorijskim prostorom. Što je manje računarsko vreme potrošeno, i što je manji zahtev za memorijom, s tim je odgovarajući algoritam bolji. Isti cilj je istaknut prilikom korišćenja particioniranja prema Bendersu. Konkretno iskustvo na računaru postoji sa celobrojno — mešovitim problemima i pokazuje porast efikasnost tih algoritama, ako se koristi Bendersovo particioniranje.
Analiziranje mogućnosti povećanja efikasnosti ostalih algoritama korišćenjem Bendersovog particioniranja zahteva daljnja istraživanja.

Resummee

Die Bedeutung des Gebrauchs der Partitionierung mit besonderen Hinblick auf die vollzähliggemischten Probleme

Wenn man über die Effektivität der Algoritmen zur Lösung der Probleme der mathematischen Programmierung spricht, dann denkt man zunächst auf die Frage, ob die Algoritmen zur Programmierung der Rechenmaschine entsprechen. Die Effektivität der Algoritmen wird auf Grund der verwendeten Rechenzeit und der zur Bearbeitung gebrauchten Zeit, festgestellt. Je weniger man Rechenzeit braucht und je weniger die Nötigkeit der Memorien ist, desto Besser ist das Algoritmus. Das selbe Ziel ist bei dem Gebrauch der Partitionierung laut Benders vorgelegt. Es gibt konkrete Erfahrungen der Rechenmaschine mit vollzählig-gemischten Problemen und es ist die Erhöhung der Effektivität der Algoritmen festzustellen, wenn man die Partitionierung von Benders benützt.
Analyse der möglichen Vergrösserung der Effektivität anderer Algoritmen, durch die Nützung der Partitionierung von Benders, ist noch zu untersuchen.