

## A raktárkészlet figyelembevétele a Gilmore—Gomory módszernél

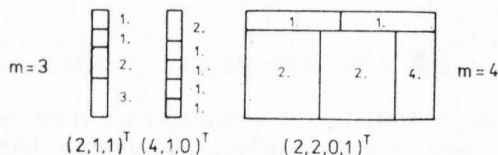
A leszabási (másképp: darabolási) feladat elsősorban üveg-, fém-, textil-, papír-, műanyagipari stb. üzemekben jelentkezik, ahol az alapanyagokat, félkésztermékeket a megadott méretű és darabszámú rendelésekre kell felválni. A leszabási feladat visszavezetése lineáris programozási feladatra közismert. A speciális feladat megoldására széles körben terjedt el P. C. GILMORE és R. E. GOMORY módszere, amely szép példája a kétfázisú módosított duális szimplex módszer gyakorlati alkalmazásának. Dolgozatunkban a szükséges alapfogalmak ismertetése után a Gilmore—Gomory módszer egy lehetséges, a raktárkészletet is figyelembe vevő módosítását mutatjuk be.

### 1. A leszabás lineáris programozási feladata

A leszabási feladatokat a felhasznált alapanyagok dimenzióinak megfelelően is csoportosíthatjuk. *Egydimenziós* a feladat pl. csövek, rudak, szalagok, stb. szétvágásakor, amikor az alapanyagoknak és az igényelt rendeléseknek csak az egyik mérete domináns. *Kétdimenziós* a darabolási feladat pl. üvegtáblák, fémlemezek, bútorlapok stb. leszabásakor, míg *háromdimenziós* feladatnak tekinthető a csomagolási, térfogatkitöltési stb. probléma.

Tegyük fel, hogy *adott* azoknak az alapanyagoknak a mérete, mennyisége és költsége, amelyek feldarabolásával kell az ismert méretű és tételszámú (igényelt darabszámú) *rendeléseket* megkapnunk. *Szabásmintán* (szabásképen, szabásrajzon, leszabási változaton stb.) a rendelések egy részének nem-átfedő elrendezését értjük az alapanyagon. A *darabolási feladat*: olyan szabásképek sorozatának a megadása, amelyek alkalmazása esetén minden rendeltést a tételszámának megfelelően kielégítünk, és a felhasznált alapanyagok összköltsége minimális.

A leszabási feladatot egészértékű lineáris programozási feladatra (a továbbiakban: ILP) vezethetjük vissza szabás-vektorok segítségével. Legyen a rendelések sorrendje rögzített és a számuk:  $m$ . Tegyük fel, hogy ezekből a rendelésekből összeállítható összes leszabási változatot ismerjük, a számuk:  $n$ . Ha egy szabásképen belül a rendelések sorrendje, elhelyezkedése az optimalizálás szempontjából tetszőleges, akkor jellemezhetjük a szabásképet egy  $m$ -dimenziós oszlop-



1. ábra. Példák egy- és kétdimenziós szabásmintákra

vektorral, amelynek  $i$ -edik komponense azt mutatja, hogy az  $i$ -edik rendelésből hány szerepel ebben a szabásmintában. Ezeket az oszlopvektorokat nevezzük *szabás-vektoroknak* (természetesen nem-negatív, egész számokból álló vektorok, lásd *I. ábra*).

A szabás-vektorok segítségével a darabolási feladat optimális megoldását a következő ILP szolgáltatja:

$$\begin{aligned}
 & x_j \geq 0, \quad x_j = \text{egész} \quad (j = 1, 2, \dots, n) \\
 & \sum_{j=1}^n a_{ij} x_j \geq N_i \quad (i = 1, 2, \dots, m) \\
 (1) \quad & \sum_{j=1}^n \delta_{kj} x_j \leq Q_k \quad (k = 1, 2, \dots, p) \\
 & \sum_{j=1}^n c_j x_j \rightarrow \min,
 \end{aligned}$$

ahol  $n$  a szabás-vektorok száma,

$m$  az eltérő méretű rendelések száma,

$p$  az alapanyag fajták száma,

$N_i$  az  $i$ -edik rendelés tételszáma,

$Q_k$  a  $k$ -adik alapanyag raktári készlete (darabszáma),

$a_{ij}$  azt mutatja, hogy az  $i$ -edik rendelésből a  $j$ -edik szabásképben hány szerepel (a  $j$ -edik szabáskép szabás-vektorának  $i$ -edik komponense),

$\delta_{kj} = \begin{cases} 1 & \text{ha a } j\text{-edik szabáskép a } k\text{-adik alapanyagból készült,} \\ 0 & \text{egyébként,} \end{cases}$

$c_j$  a  $j$ -edik szabáskép alapanyagának a költsége,

$x_j$  változó azt mutatja, hogy a megoldásban hányszor szerepel a  $j$ -edik szabáskép.

A gyakorlatban az (1) ILP megoldása nehézségeket okoz, mivel

(i) a szabásképek száma nagyon nagy,

(ii) a feladat egészértékű.

A szabásképek számának nagyságáról tájékoztat a következő becslés [6]: egydimenziós esetben, ha 200 egység hosszúságú alapanyagokból 40 különböző hosszúságú rendelést vágunk le, amelyek hossza legalább 20 és legfeljebb 80 egység, akkor a leszámbeli változatok száma 10 és 100 millió között van.

Az (1) feladat megoldására két fő megközelítés a jellemző:

(i) annyi „jó” szabásképet generálva, amennyit még a rendelkezésre álló számítógép elbír, az ILP-t közvetlenül megoldják, illetve az egészértékűségtől eltekintve az (1)-ből kapott folytonos feladatot oldják meg (pl.: [16], [17]),

(ii) a Gilmore—Gomory módszert alkalmazzák (pl.: [14], [15], [19], [20]).

A továbbiakban a Gilmore—Gomory módszert (röviden: GGM) és az általunk elkészített módosítását, bővítését ismertetjük.

## 2. A Gilmore—Gomory módszer

GILMORE és GOMORY 1961-ben az egydimenziós darabolás ILP feladatának megoldására hatékony, a gyakorlatban jól alkalmazható eljárást adott [5]. Alap gondolatukban az egészértékűségtől eltekintettek és azt használták fel,

hogy módosított szimplexmódszer alkalmazásakor egy adott bázis esetén a bázisba bevonandó vektor az szabás-vektor, így csak a legmegfelelőbb szabás-vektort kell minden egyes lépés során kiválasztani. Ez a kiválasztás pedig egyszerűen visszavezethető egy hátizsák feladatra. Az induló bázismegoldást homogén szabásképekből kapták meg. Tehát alapvetően kétfázisú módosított szimplex módszert alkalmazva, hátizsák feladatok sorozatának a megoldásával a nagyméretű, folytonos lineáris programozási feladat optimális megoldását egzaktul, viszonylag kis tárígénnel gyorsan megkapták. Eredményeiket későbbi munkáikban továbbfejlesztették és a leszábási feladatoknak szinte összes jellemző vetületét, matematikai hátterét megvizsgálták [6], [7]. A speciális hátizsák feladatok megoldására pedig kidolgozták a dinamikus programozáson alapuló ún. „hátizsák függvények” elméletét [8].

### 2.1. A GGM alapgondolata

GILMORE és GOMORY a következő folytonos feladattal foglalkozott, amelyben a raktárkészlet-feltételektől eltekintett:

$$(2) \quad \begin{aligned} x_j &\geq 0 & (j = 1, 2, \dots, n) \\ \sum_{j=1}^n a_{ij} x_j &\geq N_i & (i = 1, 2, \dots, m) \\ \sum_{j=1}^n c_j x_j &\rightarrow \min. \end{aligned}$$

Az (1) feladat gyakorlati megoldásakor két nehézséget említettünk. GILMORE és GOMORY a (2) feladat optimális megoldásának felfelé kerekítéséből kapott egészértékű megoldást (ez a közelítés természetesen szuboptimális), ami a gyakorlatban elfogadható, általában „jó” eredményt szolgáltatott.

A feladat nagy méretéből adódó problémát kétfázisú módosított duális szimplex módszerrel küszöbölték ki, így lényegében csak a bázist kell tárolni és a bázisba bevonandó vektort egy vagy több kiegészítő hátizsák feladat (továbbiakban: KP) megoldásával kapjuk meg.

Tegyük fel, hogy ismert a (2) feladatnak egy bázismegoldása, amelyben  $P_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T$  a bázisban szereplő  $j$ -edik oszlopvektor  $c_j$  árnyékköltséggel ( $j = 1, 2, \dots, m$ ),  $A = (P_1, P_2, \dots, P_m)$  és  $C = (c_1, c_2, \dots, c_m)$ .

Legyen  $P = (a_1, a_2, \dots, a_m)^T$  a bázisba bevonandó ismeretlen oszlopvektor, amelynek költsége  $c_t$  (a  $t$ -edik alapanyag költsége). A  $P$  vektor bázisba vonhatóságának feltétele a hozzá tartozó árnyékköltség pozitivitása, azaz a következő egyenlőtlenség fennállása:

$$CA^{-1}P - c_t > 0.$$

Jelöljük a  $CA^{-1}$  vektor komponenseit (amelyek a szimplex módszer táblázatában a bázisvektorok árnyékköltségei)  $\pi_1, \pi_2, \dots, \pi_m$ -el, ekkor az előzőek értelmében a bázisba bevonandó  $P$  oszlopvektort a következő feltétel szerint választjuk ki:

$$(3) \quad \sum_{i=1}^m \pi_i a_i - c_t > 0.$$

Mivel a  $P$  vektor szabás-vektor, ezért a konkrét leszabási feltételek ismeretében a gyakorlati problémák döntő részénél kiszámítható. A (3) egyenlőtlenségnek eleget tevő vektor kiszámítására a 2.2. pontban visszatérünk.

Tegyük fel, hogy a bázisba bevonandó  $P$  vektort ismerjük. A korábbi jelöléseket alkalmazva a számításokhoz használatos táblázat és a  $P$  vektor transzformáltja a következő:

$$(4) \quad \left[ \begin{array}{c|c} CA^{-1} & CA^{-1}N \\ \hline A^{-1} & A^{-1}N \end{array} \right]_{(m+1)(m+1)} P^* = \left[ \begin{array}{c|c} CA^{-1}P - c_t & \\ \hline A^{-1}P & \end{array} \right],$$

ahol  $N = (N_1, N_2, \dots, N_m)^T$ .

A módszer nem önindító, ezért szükség van egy induló bázismegoldásra, ami homogén szabásképekből áll. *Homogén egy szabáskép*, ha csak egyféle rendelés szerepel benne, azaz a szabás-vektornak csak egy komponense pozitív. Az ilyen indítás előnye az, hogy adott egy triviális bázismegoldás, aminek alapján a (4) táblázatban szereplő  $A$  diagonális mátrix inverzét könnyű kiszámítani (a főátlóban az eredeti elemek reciprokaik szerepelnek). A (4) táblázatban a jobb szélső vektor a bázisba bevihető oszlopvektor transzformáltját tartalmazza, amelyből a generálolem kiszámítható a szűk keresztmetszet elve alapján. Tehát egy induló bázismegoldás és a  $P$  vektor ismeretében a bázistranszformáció elvégezhető. A bázistranszformáció után a táblázat már az új bázisra vonatkozik és ebben kell az iterációt (a KP megoldása plusz szimplex lépés) megismételni. A (4) táblázat optimális, ha nincs olyan  $P$  vektor, amely a (3) egyenlőtlenségnek eleget tesz.

## 2.2. Egy- és kétdimenziós darabolások hátizsák feladatai

A (3) egyenlőtlenségnek eleget tevő  $P = (a_1, a_2, \dots, a_m)^T$  vektor megkeresését célszerű a megfelelő  $KP$  ( $k$ ) megoldására visszavezetni. A továbbiakban az egydimenziós darabolás és a kétdimenziós leszabás kétütemű guillotinevágásának hátizsák feladatát ismertetjük, mivel erre a két darabolási feladatra dolgoztuk ki részletesen a raktárkészlet figyelembevételéből adódó módosításokat. Az üzemi gyakorlatban ez a két darabolási típus tűnik a legfontosabbnak, a legtöbbször ezekkel találkozhatunk.

### 2.2.1. Egydimenziós darabolás hátizsák feladata

Az egydimenziós esetben legyen  $L_t$  a  $t$ -edik alapanyag hossza,  $c_t$  a költsége és  $l_i$  ( $i = 1, 2, \dots, m$ ) az egyes rendelések hossza. Ekkor a (3) egyenlőtlenségnek eleget tevő  $P$  vektort (ha létezik) a következő  $KP$  optimális megoldásaként is megkaphatjuk:

$$(5) \quad \begin{aligned} a_i &\geq 0, \quad a_i = \text{egész} \quad (i = 1, 2, \dots, m) \\ \sum_{i=1}^m l_i a_i &\leq L_t \\ \sum_{i=1}^m \pi_i a_i &\rightarrow \max. \end{aligned}$$

Amennyiben az (5) optimális megoldásának célfüggvényértéke nagyobb mint  $c_t$ , akkor az így kapott  $P$  vektor bevihető a bázisba. Természetesen a (3)-nak

eleget tevő bármelyik vektor (amely szabás-vektor) bevihető a bázisba, azonban célszerűbb a  $KP$  optimális megoldásaként kapott vektort felhasználni (amint ezt gyakorlati futtatásokkal is alátámasztották [6]). Az (5) feladatot természetesen minden alapanyaghosszra meg kell oldani.

Az (5) feladathoz további feltételként a „vágófejek” számából adódó korlátozást is csatolhatjuk, ugyanis a szabásgépek többségénél a vágófejek egyszerűen vágnak, így legfeljebb annyi vágás végezhető egyidejűleg, annyi rendelés lehet egy szabásképben, ahány vágófej van. Legyen  $R$  a vágófejek száma, ekkor a következő feltételt kell figyelembe venni:

$$(6) \quad \sum_{i=1}^m a_i \leq R.$$

Előfordulhat másmilyen, például nem-lineáris feltétel is:

$$(7) \quad \sum_{i=1}^m \text{sign } a_i \leq V,$$

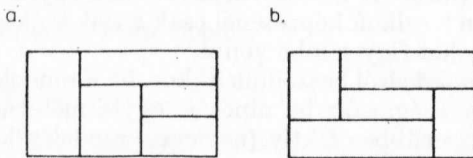
ami azt fejezi ki, hogy a csomagolás, raktározás problémái miatt legfeljebb  $V$  különböző méretű rendelés lehet egy szabásmintában (lásd [17]).

### 2.2.2. Kétdimenziós leszabások hátizsák feladatai

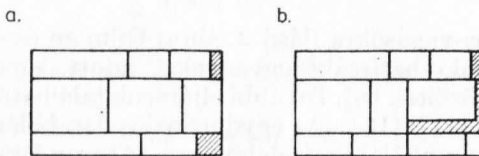
A kétdimenziós leszabások közül a legegyszerűbb a *kétütemű guillotine-vágás*, amelyet egydimenziós feladatok sorozatára vezethetünk vissza. Egy téglalapból-téglalapot darabolást *ortogonálisnak* nevezünk, ha a vágások az alapanyag valamelyik oldalára mindig merőlegesek. Egy ortogonális leszabást *guillotine-vágásnak* hívunk, ha mindig széltől-szélíg kell vágni az alapanyagot (lásd 2. ábra). A guillotine-vágás *kétütemű* (lásd 3. ábra), ha először az alapanyagot csíkokra szabjuk fel (1. ütem) és a csíkokat, mint egydimenziós alapanyagokat vágjuk tovább (2. ütem).

A guillotine-vágások egydimenziós feladatokra való visszavezetést, illetve rekurzív megközelítéseket tesznek lehetővé, mivel egy vágás után újra téglalap alakú, kisebb méretű „alapanyagot” kapunk.

Két fő lépésben határozzuk meg a  $P$  vektort, először csíkokat hozunk létre, majd a csíkok egymás mellé illesztésével a legjobb szabásképet.



2. ábra. Általános guillotine- (2a) és nem-guillotine-vágás (2b)



3. ábra. Keresztvágásos (3a) és általános (3b) kétütemű guillotine-vágás

Tegyük fel, hogy  $(L_i, W_i)$  az alapanyag mérete,  $(l_i, w_i)$  a rendelések mérete ( $i = 1, 2, \dots, m$ ), ahol  $w_1 \leq w_2 \leq \dots \leq w_m$ .

1. lépés: megoldjuk az alábbi  $KP$ -t, ezáltal a  $w_i$  szélességű legjobb csíkot megkapjuk:

$$b_j \geq 0, b_j = \text{egész} \quad (j = 1, 2, \dots, i)$$

$$\sum_{j=1}^i l_j b_j \leq L_i$$

$$\sum_{j=1}^i \pi_j b_j \rightarrow \max.$$

Legyen a célfüggvény maximuma  $\pi_i^*$ . A feladatot meg kell oldani  $i = 1, 2, \dots, m$ -re és a csík összeállítását, a  $(b_1, b_2, \dots, b_m)$  vektort minden  $i$ -re meg kell őrizni.

2. lépés: a következő  $KP$  megoldásával az 1. lépésben nyert csíkokból a legjobb szabásképet összeállítjuk:

$$d_j \geq 0, d_j = \text{egész} \quad (j = 1, 2, \dots, m)$$

$$\sum_{j=1}^m w_j d_j \leq W_i$$

$$\sum_{j=1}^m \pi_j^* d_j \rightarrow \max.$$

Ezután a  $P$  vektor a csíkok és multiplicitásaik ismeretében összeállítható azaz  $m + 1$  darab  $KP$  megoldásával elkészíthető a megfelelő szabás-vektor (A számításokat minden alapanyagra el kell végezni.)

Kétütemű guillotine-vágásnál megkülönböztetjük a keresztvágásos és az általános guillotine-vágást (lásd 3. ábra). A fentiekben az általános guillotine-vágás modellezését írtuk le. A keresztvágásos ennek egyszerűsítéséből adódik, ugyanis az 1. lépésben a csíkok képzésénél csak a csík szélességével azonos szélességű rendeléseket lehet figyelembe venni.

A *rendelés elforgathatóságáról* beszélünk akkor, ha a rendelés két mérete közül bármelyik lehet a szélesség, azaz ha nincs az egyik méretnek sem kitüntetett szerepe. Ezt a legegyszerűbben fiktív (névleges) rendelések beiktatásával modellezhetjük, legyen  $l_{m+i} = w_i$  és  $w_{m+i} = l_i$  ( $i = 1, 2, \dots, m$ ). Ekkor a  $KP$  változóinak a száma  $2m$  darab lesz. Az *alapanyag elforgathatósága* hasonlóan értelmezhető.

Általános guillotine-vágásokra (lásd 2. ábra) Gilmore és Gomory dinamikus programozáson alapuló hátizsákfüggvényeket adott, amelyek realizálására több algoritmust is közöltek [8]. További eljárások találhatóak még a következő publikációkban: [1], [11], [13]. Az egydimenziós darabolás matematikai hátterét GILMORE egyik munkájában foglalta össze [4], míg DYCKHOFF egy újszerű megközelítést vázol [2].

## 2.3. A raktárkészlet problémája a GGM-nél

Látszólag nem ütközik akadályba, hogy a raktárkészlet-feltételekkel kibővített feladatot oldjuk meg a GGM-el. A kibővített feladat az alábbi; a jelölések az (1) feladat jelöléseivel egyeznek meg:

$$(8) \quad \begin{aligned} x_j &\geq 0 & (j = 1, 2, \dots, n) \\ \sum_{j=1}^n a_{ij} x_j &\geq N_i & (i = 1, 2, \dots, m) \\ \sum_{j=1}^n \delta_{kj} x_j &\leq Q_k & (k = 1, 2, \dots, p) \\ \sum_{j=1}^n c_j x_j &\rightarrow \min. \end{aligned}$$

A (8) feladat egy induló bázisa a következő:

$$\left[ \begin{array}{c|c} A & O \\ \hline K & E \end{array} \right],$$

ahol  $A$  az induló bázismegoldáshoz tartozó szabás-vektorok mátrixa,  $E$  egységmátrix,  $K$  pedig egy 0 – 1 mátrix, amelynek elemei a (8) feladatban szereplő megfelelő  $\delta$ -k. Rendeljük az  $E$  mátrix oszlopvektoraihoz a 0 költséget, akkor a korábbi jelöléseket felhasználva az induló táblázat a következő:

$$(9) \quad \left[ \begin{array}{c|c|c} CA^{-1} & 0 & CA^{-1}N \\ \hline A^{-1} & 0 & A^{-1}N \\ \hline -KA^{-1} & E & Q - KA^{-1}N \end{array} \right],$$

ahol  $Q = (Q_1, Q_2, \dots, Q_p)^T$  a készletvektor. Ha generáloelemet nem az  $A^{-1}$  mátrix valamely sorában találunk, akkor az induló táblázat 0 mátrixa, illetve a segédvektorokhoz rendelt 0 költségvektor elromlik (segédvektort cserélünk valósi szabás-vektorra). Ezért a táblázat felső soráról csak annyit tudunk, hogy

$$\pi_1, \pi_2, \dots, \pi_m, \pi'_1, \pi'_2, \dots, \pi'_p$$

alakú, a bázisba bevonandó vektor pedig

$$(a_1, a_2, \dots, a_m, \overset{m+1}{0}, \dots, \overset{m+t}{0}, \overset{m+p}{1}, 0, \dots, 0),$$

ha a szabáskép a  $t$ -edik alapanyagra vonatkozik. Így a bázisba bevonandó  $P$  vektor árnyékköltsége

$$\sum_{i=1}^m \pi_i a_i + \pi'_t - c_t.$$

A  $KP$ -t tehát a szokásos célfüggvényre kell megoldani, így a  $KP$  „hossza” ezzel nem növekszik.

A fentiek után úgy tűnik, hogy a GGM módosítások nélkül is alkalmas a készletfeltételek kezelésére. Hogy a szakirodalom mégsem említi ilyen felhasználást, az a következőkkel magyarázható:

(i) A GGM nem önindító eljárás, ezért egy megfelelő bázismegoldás szükséges az indításához, ami szűk raktárkészletet feltételezve egy, az optimálishoz közel álló bázismegoldás ismeretét feltételezi. Másrészt tekintetbe kell venni azt is, hogy a táblázatba a bázisvektorokból álló mátrix inverzét kell beírni. Ha ez a mátrix nem diagonális, akkor az invertálás gondokat okoz, nagyméretű mátrix esetében nem biztos, hogy a későbbiekben numerikusan használható lesz ez az inverz.

(ii) Egészértékű megoldást a (8) feladat optimális megoldását felfelé kerekítve kapunk, de ekkor megsérthetjük a készletfeltételeket (gyakori lehet a túllépés).

### 3. A raktárkészlet figyelembevétele

A készletfeltételek figyelembevételekor két probléma vetődött fel. Egyrészt gondot okoz a kétfázisú módosított duális szimplex módszer induló bázismegoldásának a megadása (első fázis), másrészt a folytonos megoldás felfelé kerekítéséből nyert egészértékű megoldás megsértheti a készletfeltételeket. A fejezetben mindkét problémára adunk megoldási javaslatot.

#### 3.1. Az induló bázismegoldás elkészítése

A 2. fejezetben bemutatott GGM az induló bázismegoldást homogén szabás-minták alapján készíti el, így a felhasznált szabás-vektorok nyilvánvalóan bázist alkotnak. Az eljárás nem használható akkor, ha van olyan rendelés, amelyet nem tudunk egyféle alapanyagból kiszabni (az alapanyag szűkös raktárkészlete miatt). Ez az eset előfordulhat például akkor, amikor az alapanyagok összes felülete kevéssel haladja meg a rendelések összes felületét és kevesebb rendelés van, mint alapanyag ( $m < p$ ). Lehetséges, hogy az optimális megoldáshoz közeli induló bázismegoldást kell megkeresni, ekkor hosszú futási idővel számolhatunk és az algoritmus — a szabás-vektorok sorozatának generálása miatt — meglehetősen bonyolulttá válik. Említettük a szabás-vektorokból álló mátrix invertálásának gondját is, amely speciális szerkezetű mátrix (például háromszögmátrix), illetve iterációs eljárások felhasználását teheti szükségessé.

A fentiek miatt az induló bázismegoldás elkészítésére egyszerű ötleten alapuló eljárást választottunk, a *rendeléseket olyan részrendelésekre bontjuk*, amelyek külön-külön már egyféle alapanyagból kiszabhatók homogén szabásminták szerint. Azaz a rendeléseket „névleges” részrendelésekkel helyettesítjük, vizsgázza arra, hogy a rendelések darabszámainak összege az eredeti rendelés tételszámával egyezzen meg. Így (a számításokhoz használt táblázat méreteinek növelése árán) az induló bázismegoldás vektorainak a mátrixa már diagonális lesz (mivel homogén szabás-vektorokból áll), és a táblázat a szokott módon indítható. Megjegyezzük, hogy a táblázat méreteinek a növelése a  $KP$ -k megoldásának idősükségletét nem növeli (ami a becslések szerint egydimenziós esetben a teljes számítási idő 80%-a [15]), mert a  $KP$  hossza tulajdonképpen nem nő. Ugyanis, ha az (5)  $KP$ -nál a részrendelések mérete azonos, akkor elég közülük azt megtartani, amelynek az árnyékköltsége nagyobb, mivel ha  $l_i = l_j$  és  $\pi_i < \pi_j$ , akkor  $a_i = 0$ .

Az induló bázismegoldáshoz az összes homogén szabásképből választjuk ki a megfelelőket és a kiválasztás alapján bontjuk szét a rendeléseket részrendelé-



sekre. Természetesen ha van rá lehetőség, akkor elkerüljük a rendelések szétbontását. *Egydimenziós feladat*nál legyen az alapanyagok és a rendelések sorrendje rögzített. Defináljuk a  $D = [d_{ij}]_{j=1, p}^{i=1, m}$  szabásmátrixot és az  $E = [e_{ij}]_{j=1, p}^{i=1, m}$  fajlagos felületveszteség-mátrixot a következőképpen:

$$d_{ij} = \begin{cases} [L_j/l_i], & \text{ha } l_i \leq L_j, \\ 0 & \text{egyébként,} \end{cases}$$

$$e_{ij} = \begin{cases} (L_j - l_i d_{ij})/d_{ij}, & \text{ha } d_{ij} > 0, \\ \infty & \text{egyébként,} \end{cases}$$

ahol [ ] az egészrész jele. Ekkor a homogén szabás-vektorokra a következő feladat írható fel:

$$x_{ij} \geq 0, \quad x_{ij} = \text{egész} \quad \begin{matrix} (i = 1, 2, \dots, m; \\ j = 1, 2, \dots, p) \end{matrix}$$

$$\sum_{j=1}^p d_{ij} x_{ij} \geq N_i$$

$$(10) \quad \sum_{i=1}^m x_{ij} \leq Q_j$$

$$\sum_{i=1}^m \sum_{j=1}^p e_{ij} d_{ij} x_{ij} \rightarrow \min,$$

ahol  $x_{ij}$  mutatja azt, hogy a megfelelő homogén szabás-vektor hányszor szerepel a megoldásban. A célfüggvény most közvetlenül a felületveszteség mennyiségét minimalizálja és nem az alapanyag költségét.

Vegyük észre: ha a (10) képletekből a  $d_{ij}$  együtthatókat elhagyjuk és az egyenlőtlenségek helyett egyenlőséget írunk, akkor a közismert szállítási feladat áll előttünk. Ezért az annak megoldására alkalmazott disztribúciós kezdő megoldását szolgáltató heurisztikus eljárás egy módosításával határozzuk meg az  $x_{ij}$  értékeket a következőképpen:

$$i = 0$$

1. lépés:  $i = i + 1, x_{ij} = 0, (j = 1, 2, \dots, p).$

2. lépés:  $j = \{k \mid \min_k e_{ik}\},$

ha  $e_{ij} = \infty$ , menj a 3. lépésre.

$$e_{ij} = \infty$$

$$x_{ij} = \min_j \{Q_j, [(N_i + d_{ij} - 1)/d_{ij}]\}$$

$$Q_j = Q_j - x_{ij}$$

$$N_i = N_i - x_{ij} d_{ij}.$$

Ha  $N_i > 0$ , menj a 2. lépésre.

3. lépés: Ha  $i < m$ , menj az 1. lépésre.

Vége.

Látható, hogy az eljárás a legkisebb veszteségű homogén szabásképeket használja fel minél többször, az aktuális igény és a készlet figyelembevételével. Ezután a részrendelésekre bontás egyértelmű (az  $i$ -edik rendeltést  $\Sigma_j \text{sign } x_{ij}$

féle részre kell bontani), és az induló bázismegoldás is megvan. Így a (9) táblázat a  $D$  és  $X = [x_{ij}]$  mátrixok ismeretében feltölthető, tehát a GGM-iterációk ( $KP$  plusz bázistranszformációk) elvégezhetőek.

A fenti eljárás csaknem dimenzió-független, így *kétdimenziós feladat*nál is alkalmazható kis módosítással. Csupán a  $d_{ij}$  és az  $e_{ij}$  kiszámítása változik.

Ha az alapanyagot a rendelés nem forgatható el, akkor

$$d_{ij} = \begin{cases} \left\lceil \frac{L_j}{l_i} \right\rceil \left\lceil \frac{W_j}{w_i} \right\rceil, & \text{ha } l_i \leq L_j \text{ és } w_i \leq W_j, \\ 0 & \text{egyébként,} \end{cases}$$

elforgatható rendelés esetén pedig:

$$d_{ij} = \max \left\{ f_1 \left\lceil \frac{L_j}{l_i} \right\rceil + f_2 \left\lceil \frac{W_j}{w_i} \right\rceil \right\}$$

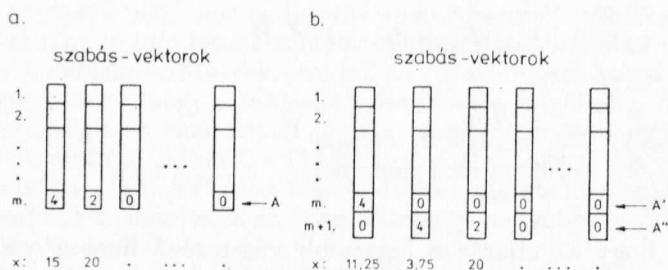
ahol  $w_i f_1 + l_i f_2 \leq W_j$   
és  $f_1, f_2 \geq 0$ , egészek,  
feltéve, hogy  $\min\{l_i, w_i\} \leq \min\{L_j, W_j\}$  és  $\max\{l_i, w_i\} \leq \max\{L_j, W_j\}$ .  
Tehát az utóbbi esetben egy  $KP$  megoldásaként adódik a homogén szabásvektor (kétféle homogén csíkot is figyelembe veszünk egy szabásképpben).

Az  $e_{ij}$  pedig a következő:

$$e_{ij} = \begin{cases} (L_j W_j - l_i w_i d_{ij}) / d_{ij}, & \text{ha } d_{ij} > 0, \\ \infty & \text{egyébként.} \end{cases}$$

A fenti eljárás a lehetséges megoldások halmazát is előzetesen teszteli, így ha az eljárás nem ad bázismegoldást, akkor célszerű valamelyik készletfeltétel lazítása, feloldása.

A rendeléseknek részrendelésekre bontásával a (8)  $LP$  feltételeinek a száma megnő, és maguk a feltételek is megváltoznak. Első látásra úgy tűnik, hogy az egzakt GGM ezáltal heurisztikussá válik. Valójában a darabolási feladatnak és a szabásmintáknak a jellemzői miatt a részrendelések bevezetése után is egzakt marad a módszer. Ez a legegyszerűbben azzal bizonyítható, ha belátjuk, hogy az eredeti  $LP$  optimális megoldása egyben megoldása a részrendelésekre felírt  $LP$ -nek is. A visszavezetést egyszerű példa segítségével szemléltetjük (mellőzve a nehézkes formalizmust): tegyük fel, hogy az  $A$  rendelés (amelyből 100 db-ot kértek) az eredeti feladat optimális megoldásának két szabásvektorában szerepel, az egyik szabásmintát 15-ször, míg a másikat 20-szor kell ki szabni



4. ábra. Az eredeti feladat optimális megoldásának (a) és a részrendelésekre bontott feladat egy lehetséges megoldásának (b) szabásvektorai és multipllicitásai

Tegyük fel továbbá, hogy a rendelkezések közül *csak a  $A$  rendelkezést* kellett részrendelesekre bontani, a részrendelesek ( $A'$  és  $A''$ ) tételszámai 45 és 55. Ekkor az eredeti feladat optimális megoldása alapján a 4. ábrán vázolt lehetséges megoldása készíthető el a részrendelesekre bontott feladatnak (ahol  $A$  legyen az eredeti feladat  $m$ -edik rendelkezése, míg  $A'$  az  $m$ -edik,  $A''$  az  $m + 1$ -edik a részrendelesekre bontott feladatban).

A fordított visszavezetés is hasonlóan triviális a szabás-vektorok és multiplícitásaik felbontása alapján. Így a két feladat ekvivalens, optimális megoldásuk megegyezik. A gyakorlati megoldások viszont eltérhetnek az alkalmazott módszerek különbözőségei és a kerekítési hibák miatt.

### 3.2. Virtuális raktárkészlet bevezetése

A GGM a (8) feladat folytonos megoldását felfelé kerekíti, így gyakran előfordulhat, hogy a készletfeltételt megsértik. A kerekítésből adódó hibák, túllépések kiküszöbölésére a következő egyszerű gondolat kínálkozik. A (8) feladat készletfeltételeinek jobb oldalát a szükséges mértékben csökkentjük, legyen:

$$(11) \quad Q'_k = Q_k - \min \{m + p, \sum_{j=1}^n \delta_{kj}\}, \quad (k = 1, 2, \dots, p).$$

Ekkor biztos, hogy a készletfeltételekre a felfelé kerekítés után is teljesül az egyenlőtlenség (figyelembe vettük, hogy legfeljebb  $m + p$  szabásképet tartalmaz egy bázismegoldás). Komoly problémát okoz viszont az, hogy egyidejűleg csak a bázisban szereplő szabás-vektorok ismertek, nem a teljes feltételrendszer. Másrészt nem lenne célszerű az összes szabás-vektort a (11) képletnél figyelembe venni; elég, ha a bázist alkotó szabás-vektorok alapanyagának készletét csökkentjük, változtatjuk a bázisvektor-cserék alkalmából, ezért legyen:

$$(12) \quad Q'_k = Q_k - \min \{m + p, \sum_{j=1}^m \delta'_{kj}\}, \quad (k = 1, 2, \dots, p),$$

ahol  $\delta'_{kj}$  csak a bázist alkotó szabás-vektorokhoz tartozik.

A fentieknek megfelelően *virtuális raktárkészletet* ( $Q'_k$ ) hozunk létre minden alapanyagból, és minden bázisvektor-csere esetén a bázisból kikerülő szabásvektor alapanyagának virtuális készlete 1-el nő, míg a bázisba bekerülő szabásvektoré 1-el csökken.

Ez a változtatás azt jelenti, hogy a (8) feladatban a feltételek minden bázisvektor-csere alkalmából megváltozhatnak (akkor maradnak változatlanok, ha a belépő és a kilépő vektor ugyanolyan alapanyagot használ). A (9) táblázatban gondos programozással végigvihető az elképzelésünk, azonban a virtuális raktárkészlet alkalmazása egyéb módosításokat is maga után von.

### 3.3. Egyéb módosítások

A bázisba bevihető szabás-vektort úgy keressük meg, hogy minden alapanyagra a  $KP$ -k megoldásával megállapítjuk a „legjobb” szabás-vektort. A GGM az árnyékköltségeket összehasonlítva választja ki ezek közül a legelőnyösebb szabás-vektort, amit azután bevisz a bázisba.

A legelőnyösebb árnyékköltségű szabásvektor bázisba vitele helyett mi a következőket végezzük el: mindegyik alapanyagra imitáljuk a bázistranszfor-

mációt és azt a szabás-vektort vonjuk be a bázisba, amelyre a célfüggvény a legnagyobb javulást mutatja. Azaz a (9) táblázat jobb szélső oszlopán végrehajtjuk a virtuális raktárkészlet megkövetelte változtatást (a bázisba bevonandó alapanyag készlete 1-el csökken) és főelem-kiválasztással megállapítjuk a bázisból kikerülő vektort (az alapanyagának készletét 1-el növeljük). Végrehajtjuk a bázistranszformációt, de csak a jobb oldali oszlopot és a célfüggvény értékét számoljuk ki. Miután ezt mindegyik alapanyagra elvégezzük, azt a vektort visszük be a bázisba, amelyik a legnagyobb javulást eredményezi a célfüggvényénél. Ez azért szükséges, mivel a feltételrendszer jobb oldalának folytonos változtatása miatt az árnyékköltségek vizsgálata nem elég a javulás megállapításához, a próbafuttatások során előfordult a célfüggvény értékének „hullámszáma”, a változás nem volt monoton.

Az iterációs lépések csökkentésére a szokásos módon küszöbszámot érdemes használni ( $\varepsilon > 0$ ), amelynek nagysága a gyakorlati feltételektől függ. Az aktuális bázismegoldást optimálisnak fogadjuk el, ha a célfüggvény értéke a megadott küszöbszámnál kevesebbet javul. Ezáltal a szimplex módszernél esetleg fellépő degeneráció is kiszűrhető.

A virtuális raktárkészlet bevezetése a módszert heurisztikussá teszi. Megjegyezzük azonban, hogy így egyrészt egyszerű eszközökkel elkerülhetjük a raktárkészlet-feltételek megsértését (mellőzhetjük az egészértékű megoldást kereső bonyolultabb eljárásokat), amely a folytonos feladat megoldásának egészértékűvé kerekítésekor fordulhat elő. Másrészt a szakirodalom szerint a GGM-t akkor célszerű használni, ha a rendelkezés és alapanyagok tételszáma „elég nagy” (ami azt jelenti, hogy a megoldásban a szabás-vektorok multiplicitása legalább kétjegyű szám), ekkor a raktárkészlet minimális csökkentése nem befolyásolja lényegesen a hulladékot (mivel  $Q_k \approx Q'_k$ ). Ezért ebben az esetben a virtuális raktárkészlet nem okoz számottevő eltérést az optimális megoldástól.

#### 4. A GGM módosításának számítógépes realizálása

Az előző fejezetben bemutatott módosításokat egydimenziós darabolásokra és kétütemű kétdimenziós guillotine-vágásokra kipróbáltuk, az algoritmusok számítógépes futtatását elvégeztük. A gépidő- és a társzükséglet természetesen a leszábási feladat konkrét jellemzőitől, a felhasznált számítógéptől és a programrendszer felépítésétől függ. Becsléseink szerint az eredeti GGM-hez képest, ahol a raktári feltételeket nem vesszük figyelembe, kb. 10–20%-kal nő meg a gépidő. A növekedés jelentős részét a bázistranszformáció alapanyagokénti imitálása okozza, míg az induló bázismegoldás kiszámításához szükséges idő elenyésző. A tárigény a megoldandó feladat függvénye, a részrendelésekre bontás mértéke előre nem adható meg, azonban becslésünk szerint egy 32 Kszó tárcapacitású számítógéppel is kezelhető legalább 70 (rész-)rendelés. Az általunk megvalósított egydimenziós programrendszer helyfoglalása a bináris kódon túl még  $2(m' + p^2) + m'p + 12m'$  szó (1 szó = 1 egész szám), ahol  $m'$  a részrendelések,  $p$  pedig az alapanyagok száma.

Az egydimenziós és a kétdimenziós darabolásra készített programrendszerek legfontosabb jellemzőit röviden ismertetjük. Mindkét rendszer HP-1000 számítógépre készült (32 Kszó = 64 Kbyte tárcapacitású) FORTRAN-ban.

Megítélésünk szerint hajlékony mágneslemezzel rendelkező személyi számítógépekre is elkészíthetők a rendszerek, habár nagyon hosszú futási időre kell számítani (a programrendszereket 1982-ben, a személyi számítógépek hazai megjelenése előtt készítettük el).

Az *egydimenziós* darabolásnál az alapanyagok és rendelések jellemzőin kívül (ahol  $p \leq 10$  és  $m' \leq 70$ ) a használható vágófejek maximális számát és a konvergencia ellenőrzését biztosító paramétert ( $\epsilon$ ) is megadjuk az inputban. Az (5) *KP* tehát a (6) vágófej-feltétellel kibővül. A *KP* megoldására a [6] publikációban ismertetett eljárást használjuk, amely a vektorok lexicografikus elrendezésén alapszik. Az eljárás előnye, hogy az összes alapanyagra szimultán oldja meg a *KP*-kat, kihasználva, hogy csak a feltételek jobb oldalában térnek el egymástól. További előnye, hogy a vágófejekből adódó feltételt egyszerű figyelembe venni, és a program helyigénye kicsiny ( $L_j$ -től független, 2 mp a tárigény). Ezt a leszámoló algoritmust használja több programrendszer, például [3], [14]. Megjegyezzük, hogy a (7) nem-lineáris feltételt az eljárás nem veszi figyelembe, viszont egy-egy rendelés számának szabásképben való korlátozását be lehet építeni [10].

*Kétdimenziós* leszabásnál az alapanyagok és a rendelések jellemzői között meg kell adni (ahol  $p \leq 10$  és  $m' \leq 50$ ), hogy elforgathatók-e. Mivel kétütemű guillotine-vágásról van szó, ezért lehet keresztvágásos vagy általános a leszabás (lásd 3. ábra), amit szintén jelezni kell. A vágófejek maximális számát és a konvergencia-faktort, illetve az alapanyag szélén kötelezően levágandó technológiai selejtsík szélességét is meg lehet adni. A *KP*-k megoldására ugyancsak a lexicografikus rendezésen alapuló leszámoló algoritmust használjuk. (A szabásmintákat a számítógép megközelítőleg méretarányosan kirajzolja.) Megjegyezzük, hogy a szabás-vektorokkal együtt a csik-vektorokat is tárolni kell (a reprodukálhatóság végett), a nagy mennyiségű adattömeget célszerű közvetlen elérésű háttértárolón elhelyezni (például mágneslemezen).

A témakör szakirodalmában a GGM alkalmazásáról igen sokszor számolnak be. Azonban olyan publikációt és futtatási eredményeket nem ismerünk, amely a raktárkészletet, mint korlátozó feltételt veszi figyelembe; így az általunk alkalmazott módosításokat nem tudjuk teljeskörűen összevetni más eredményekkel. MARCONI ugyan felveti a raktárkészlet okozta problémát, de megoldást nem ad rá szűkebb körben publikált tanulmányában [20]. A módszer és a módosítások illusztrációjaként a [17] dolgozatban található leszabási feladatra — amely húzott síküveg feldarabolását mutatja be — közöljük a futtatási eredményeket. A leszabás általános feltételének a következőket vetjük: kétütemű guillotine-vágás, a vágófejek száma 8, a technológiai selejtsík 7 cm, a konvergencia-faktor 1,0. Az előbbi általános leszabási feltételeken belül

#### 1. táblázat

Az alapanyagok jellemzői, input adatok

Azono- sító	Méretek	Készlet	Ár
T1	200×200	2000	4,0
T2	200×220	2000	4,4
T3	200×240	2000	4,8

## 2. táblázat

A rendelkezések két csoportjának jellemzői (minden rendelés elforgatható)

Azono- sító	Méretek	Tételszám	Azono- sító	Méretek	Tételszám
A1	134×74	420	B1	152×80	220
A2	50×74	207	B2	184×84	700
A3	134×132	300	B3	100×100	200
A4	50×132	150	B4	162×78	300
A5	132×102	204	B5	130×82	150
A6	50×102	170	B6	134×74	280
A7	132×132	204	B7	50×74	138
A8	86×132	204	B8	134×132	200
A9	84×84	204	B9	50×132	100
A10	134×44	204	B10	132×102	136
A11	150×134	80	B11	132×132	136
A12	96×85	80	B12	86×132	136
A13	82×82	80	B13	84×84	136
A14	150×44	80	B14	134×44	136

1. csoport

2. csoport

több eltérő leszabási változatra számoltuk ki az optimális szabás-vektorokat. A bemenő adatok az 1. és a 2. táblázatban találhatóak meg (minden adatot cm-ben adunk meg). A 3. táblázat az eltérő leszabási feltételek mellett az optimális darabolások hulladékát (a felületvesztesség %-át) tartalmazza, ahol:

$$\text{hulladék \%} = \frac{\text{felvágott alaptáblák összfelülete} - \text{rendelések összfelülete}}{\text{felvágott alaptáblák összfelülete}} \times 100.$$

Mivel a technológiai selejtesík, amelyet az alaptábla szélén kötelező levágni, az egyes darabolási feltételek szemléltető hatását eltorzíthatja, ezért a technológiai selejt nélkül is megadjuk az egyes hulladékokat, amely a táblázatban az adott csoport alsó sorában található.

A [17] dolgozatban a leszabás konkrét feltételei a következők: az alapanyag nem forgatható el, keresztvágásos kétütemű guillotine-vágás lehetséges (3. a ábra) és csak az alapanyag szélességével párhuzamosan vágathatók a csíkok (az alaptáblák 200 cm-es fix méretét *szélességnek*, a változó méretet pedig *hosszúságnak* nevezzük). A 3. táblázat 1. oszlopában a hivatkozott cikk eredményei, a 2. oszlopban ugyanazon feltételek mellett a GGM-el kiszámolt szabás-vektorok hulladékai találhatóak, így a két szomszédos oszlopban szereplő adatok összevethetőek.

A [17] dolgozattal összevethető, azonos leszabási feltételek alapján készült optimális szabás-vektorokat mindkét csoportra a 4. táblázatban közöljük. A szabás-vektor  $i$ -edik komponense a beolvasás sorrendjében  $i$ -edik rendeléshez tartozik, a nullákat nem tüntettük fel, a szabás-vektor itt sorvektor. A 3. táblázatból látható, hogy minimális az eltérés a két módszer között, ami a kerekítések miatt és a konvergencia-faktor alkalmazásából adódhat. [A (7) feltételt mi nem vettük figyelembe, viszont nem is sértettük meg.] A GGM előnyére írható, hogy kisszámítógépen közel 50 rendelés leszabását lehet optimalizálni, míg a [17]-ben nagyszámítógépen csak 20 rendelését (ez utóbbi egy nagymé-

3. táblázat

A hulladék %-os alakulása a kétütemű guillotine-vágás eltérő szabási feltételei esetén

	Az alaptábla <i>nem</i> forgatható el					Elforgatható alaptábla	
	Keresztvágásos			Általános		Keresztvágásos	Általános
	Szélességgel párhuzamos csík		Hosszal párh. csík	Szélességgel párh. csík	Hosszal párh. csík		
	1.	2.	3.	4.	5.	6.	7.
1. csoport	22,02	22,39	24,35	21,59	22,79	21,12	19,46
	16,61	17,03	19,07	16,22	17,46	15,79	14,07
2. csoport	23,65	23,57	28,97	22,14	23,48	22,92	20,65
	18,23	18,14	23,86	16,43	18,10	17,53	15,13

Megjegyzés: a) Csoportonként az alsó érték a technológiai veszteség nélküli szabási hulladékot tartalmazza.

b) Az 1. oszlopban a [17] dolgozatban szereplő optimális szabásvektorok hulladéka található, míg a 2. oszlopban a megfelelő GGM-eredmények.

4. táblázat

A két csoport optimális szabás-vektorai

Alap-tábla	Hány db	Szabás-vektorok													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
T2	34	1				1			1						
T2	3	1	1		1			1							
T2	96	1		1											
T2	85					2	2								
T3	16							1	1						
T2	204	1	1	1							1				
T2	38	1						1							
T3	52								3						
T3	102				1			1		2					
T2	5	1			1			1							
T3	40	1										1			1
T3	40				1			1					2		
T3	40											1		2	1

1. csoport. Az optimális szabás-vektorok hulladéka: 22,39%

T1	34		1	1											
T1	299		2												
T2	65				1			1							1
T2	136				1						1		1		
T3	47			1		2									
T3	51			1				1	1						
T3	69		1				2	2							
T2	28				1	2									
T2	49	1								1		1			
T2	71				1		2								1
T1	68			1										2	
T2	84	1							1						
T2	87	1									1				

2. csoport. Az optimális szabás-vektorok hulladéka: 23,57%

tű ILP-t generál és old meg). Az egyes darabolási jellemzők hatását a hulladék alakulására a 3. táblázat szemléletesen mutatja.

Végeztünk futtatásokat szűk raktárkészlet esetében is. Például az 1. csoportnál, ha  $T_1 = 400$ ,  $T_2 = 400$  és  $T_3 = 400$ , akkor a 3. táblázat specifikációi alapján a 2. oszlopnak 24,12%, a 3. oszlopnak 33,79%, míg az 5. oszlopnak 27,60% felel meg. A hulladék jelentősen megnő, azonban ez a növekedés nem egyedül a raktárkészlet szűkösségétől függ (habár az a döntő), hanem a rendelések méreteinek „összeilleszthetőségétől” is.

\* \* \*

Dolgozatunkban a GGM egy gyakorlati szempontú, számítógépre egyszerűen adaptálható bővítését, módosítását ismertettük, amelynek segítségével az alapanyagok raktári készletét is korrekt módon figyelembe lehet venni. Kedvező esetben nincs szükség az alapanyag készletezésére, mivel az optimális leszabáshoz szükséges mennyiségben és méretben megrendelve a darabolás megkezdéséig megérkezik. Azonban előfordulhat, hogy a termelés ütemtelensége, a megrendelések hosszú átfutási ideje, az előre nehezen becsülhető felhasználás stb. miatt a vállalat nagy biztonsági készletet hoz létre, és ebből a raktári készletből kell az üzemi igényeket kielégíteni. Másrészt egyes alapanyagoknál technológiai okok miatt van szükség a raktározásra, például húzott sfküvegnél csak a húzás után dől el az alaptábla minősége, ezért jelentős mennyiségű raktárkészlet is felhalmozódhat olyan minőségű üvegtáblákból, amelyre az adott pillanatban nincs piaci igény.

A módszer alkalmazásakor felvetődő szervezési munkákra és egyéb kapcsolódó feladatokra nem térünk ki, mivel a célunk a GGM, egy eszközként funkcionáló eljárás, módosításának ismertetése. A leszabási feladattal foglalkozó szakirodalom igen bőséges, az üzemi feltételeket messzemenően figyelembe vevő (főleg heurisztikus) eljárások tömegét dolgozták ki (lásd [18]). A darabolási feladatok iránt érdeklődőknek GILMORE és GOMORY munkáin kívül [4—8] első sorban GOLDEN elméleti [9] és HINXMAN alkalmazói szempontú [12] áttekintését ajánljuk.

(Beérkezett: 1983. szeptember 13-án.)

## IRODALOM

1. CHRISTOFIEDS, N.—WHITLOCK, C.: An algorithm for two-dimensional cutting problems. *Operations Research* 25, 1977 (30—44).
2. DYCKHOFF, H.: A new linear programming approach to the cutting stock problem. *Operations Research* 29, 1981 (1092—1104).
3. DYSON, R. G.—GREGORY, A. S.: The cutting stock problem in the flat glass industry. *Operational Res. Quart.* 25, 1974 (41—53).
4. GILMORE, P. C.: Cutting stock, linear programming, knapsacking, dynamic programming and integer programming, some interconnections. *Annals of Discrete Math.* 4, 1979 (217—235).
5. GILMORE, P. C.—GOMORY, R. E.: A linear programming approach to the cutting stock problem. *Operations Research* 9, 1961 (849—859).
6. Op. cit. Part II. *Operations Research* 11, 1963 (863—888).
7. GILMORE, P. C.—GOMORY, R. E.: Multistage cutting stock problems of two and more dimensions. *Operations Research* 13, 1965 (94—120).
8. GILMORE, P. C.—GOMORY, R. E.: The theory and computation of knapsack functions. *Operations Research* 14, 1966 (1045—1074).



9. GOLDEN, B. L.: Approaches to the cutting stock problem. *AIIE Transactions* 8, 1976 (265—274).
10. HAESSLER, R. W.: A note on computational modifications to the Gilmore-Gomory cutting stock algorithm. *Operations Research* 28, 1980 (1001—1004).
11. HERZ, J. C.: Recursive computational procedure for two-dimensional stock cutting. *IBM Journal of Res. and Dev.* 16, 1972 (462—469).
12. HINXMAN, A. I.: The trim-loss and assortment problems: a survey. *European Journal of Operational Res.* 5, 1980 (8—18).
13. HODGSON, T. J.: A combined approach to the pallet loading problem. *IIE Transactions* 14, 1982 (175—182).
14. JOHNSTON, D.—BOURKE, S. B.: The development of computer programs for rools deekle filling. *APPITA* 26, 1973 (444—448).
15. KUUTTI, M.—VOUTILAINEN, R.: *A discrete optimization approach to the pattern analysis of the cutting stock problem*. Paper presented at EURO III. Conference, Amsterdam (Netherlands), 1979.
16. LAMPL, T.: Anyagleszabási tervek készítése és programozása matematikai eszközökkel *Operációkutatási Értékezés*, (SZÁMOK), 1971.
17. LENGYEL, I.—KUBA, A.: Sfküveg szabásának optimalizálása (I) *Sigma* 14, 1981 (169—190).
18. LENGYEL, I.: *A darabolási feladatok és a megoldásukra alkalmazott módszerek áttekintése* Egyetemi doktori értekezés, 1982.
19. MADSEN, Oli. B. G.: Glass cutting in a small firm. *Mathematical Programming* 17, 1979 (85—90).
20. MARCONI, R.: *Generalization of a mathematical procedure for the optimal solution of two-dimensional cutting problems*. Technical reports No. 7. Centri Studi IBM of Pisa, 1970.

## THE CONSIDERATION OF STOCK WITHIN THE GILMORE—GOMORY METHOD

In industrial shops the demand for solving the cutting problem frequently arises when the given raw materials and intermediary products have to be cut into a given number of pieces of a specified size. For the solution of this problem the method worked out by P. C. Gilmore and R. E. Gomory is widely applied which solves the linear programming problem of cutting with the aid of a modified dual simplex method so that the vector to be drawn into the basis is obtained as a solution of a knapsack problem. In this paper an extension and modification of the Gilmore—Gomory method is presented so that also the constraints deriving from the stock of the materials can be taken into account. Two problems emerged: on the one hand the finding of an initial basic solution itself and, on the other hand the integer solution obtained by rounding-off the optimum solution of the continuous linear programming problem may exceed the stock. For the solution of both problems a simple, easily computerized procedure has been prepared. The computerization of the procedures is briefly reviewed.

## УЧЕТ СКЛАДСКИХ ЗАПАСОВ С ПОМОЩЬЮ МЕТОДА ЖИЛМОРА-ГОМОРИ

На промышленных предприятиях часто возникает потребность в решении задач отсечения, когда необходимо разделить имеющиеся материалы и полуфабрикаты в соответствии с заданными размерами и количеством. Для решения данной задачи очень широко применяется метод, разработанный П. С. Жилмором и Р. Е. Гомори. Когда задача отсечения решается с помощью модифицированного двойственного симплексного метода таким образом, что вводимый в базис вектор получают в результате решения задачи о ранце.

В данной статье излагается такое обобщение метода Жилмора-Гомори, с помощью которого можно учесть ограничения на складские запасы. При этом возникли две проблемы: во-первых, выбор исходного базисного решения, во-вторых, оптимальное решение задачи линейного программирования при округлении для получения целочисленного решения может превысить складской запас. В обоих случаях авторами разработан простой и легко адаптируемый на ЭВМ метод, а также кратко излагается их вычислительный алгоритм.