Lengyel Imre

Anyagleszabási problémák hátizsákfeladatainak egy gyakorlati célú általánosítása

1. Bevezetés

Az üzemekben gyakran felmerülnek anyagleszabási (darabolási, vágási, nyírási stb.) problémák, amikor bizonyos szabványos anyagokból az igényeknek megfelelően kisebb méretűeket kell kiszabni. Cél olyan szabásminták megadása, amelyek alkalmazása esetén a felhasznált anyagok összköltsége minimális. Az anyagleszabási probléma megoldására a közismert lineáris programozási feladatra való visszavezetés mellett számos heurisztikus eljárást is kidolgoztak (lásd Dowsland [4], Dyckhoff, Kruse, Abel és Gal [5], Golden [12], Hinxman [19], Lengyel [27] összefoglalóját). Az alkalmazott módszerek közül kiemelhető két, a gyakorlatban is legtöbbet felhasznált megközelítés:

1. A Gilmore–Gomory módszer (röviden: GGM), amely a darabolás lineáris programozási feladatát módosított szimplexmódszerrel oldja meg úgy, hogy a bázisba bevonandó vektort hátizsákfeladatok megoldásával állítja elő ([7–11], [22]).
2. Heurisztikus szekvenciális eljárás (kiürítő eljárás), amely valamilyen szempontok szerint „legjobb" szabásmintát állít elő, ezt a szabásmintát a lehető legtöbbször alkalmazza, majd a megmaradt alapanyagokra és idomokra állít elő „legjobb" szabásmintát stb. (pl. [1], [15], [18], [30], [32]). A „legjobb" szabásmintát általában hátizsákfeladatnak egzakt vagy közelítő megoldásából konstruálják.

Mindkét esetben a hátizsákfeladat megfogalmazásának (amely a szabásmintával szemben támasztott technológiai, műszaki, szervezési stb. elvárásokat modellezi) és a feladat kevés időt igénylő, gyors megoldásának kiemelkedő fontossága van.

Dolgozatunkban az anyagleszabás gyakorlati körülményeinek újszerű modellezését, a hátizsákfeladat általánosításaként adódó feladatot és a feladat megoldására kidolgozott eljárást ismertetjük. Az anyagleszabási problémák közül az egydimenziós darabolás (papírtekercsek, rudak, csövek, szalagok stb. szétvágása) hátizsákfeladatának általánosításával foglalkozunk részletesen, amely mind a GGM-nél, mind a heurisztikus szekvenciális eljárásoknál ugyanaz. Kétdimenziós két-ütemű guillotine-vágásra, ahol a kétdimenziós leszabás egydimenziós darabolások sorozatára vezethető vissza, csak röviden térünk ki.

A dolgozat 2. fejezetében a leszabás üzemi (technológiai, műszaki, szervezési stb.) feltételeinek modellezését és a hátizsákfeladat általánosításaként kapott feladatot ismertetjük. A 3. fejezetben bemutatjuk a feladat megoldására kidolgozott eljárást, a 4. fejezetben a gyakorlatban alkalmazott heurisztikus módosításokra térünk ki. Az 5. fejezet a számítógépes futtatások eredményeit tartalmazza.

2. Az egydimenziós leszabás általános feltételei

Egydimenziós leszabásnál legyen L a rendelkezésre álló alapanyag hossza, a_i $(i = 1, 2, \ldots, m)$ a kért idomok méretei. Ekkor a hátizsákfeladat a következőképpen írható fel:

$$0 \leq x_i \text{ egész } (i = 1, 2, \ldots, m) \tag{1}$$

$$\sum_{i=1}^{m} a_i x_i \leq L \tag{2}$$

$$\sum_{i=1}^{m} c_i x_i \to \max. \tag{3}$$

A szabásmintát a hátizsákfeladat megoldása mutatja, ahol x_i azt jelzi, hogy az i -edik idom hányszor szerepel a szabásmintában. A (2) feltétel azt fejezi ki, hogy az alapanyag hosszánál nem lehet nagyobb a belőle levágandó idomok összhossza, azaz nem-átfedő lefedésről van szó. A (3) célfüggvény c_i együtthatói $(c_i > o)$ a GGM-nél a megfelelő bázisvektorok árnyékárai, heurisztikus módszereknél pedig valamilyen pozitív súlyszámok (lehet az idomok hossza is, ekkor megkapjuk a legkisebb hulladékú szabásmintát).

Az (1—3) feladat megoldásaként kapott vektorok több esetben nem elégítik ki a szabásmintákkal szemben támasztott üzemi elvárásokat. Például, ha a szabászgép vágófejei (kései, fűrészei) egyszerre vágnak, akkor a vágófejek maximális számát is figyelembe kell venni. A levágott idomok válogatásának, csomagolásának, nyilvántartásának nehézségei is a feladat módosítását igényelhetik stb. Mivel a gyakorlatban a leszabás jó szervezhetősége alapvető, sőt a legfontosabb szempont, ezért a fenti hátizsákfeladat megoldásából konstruált szabásminták alkalmazása sokszor nehezen vihető végbe. Az egyik összefoglalóban [5] a szerzők 34 gyakorlati alkalmazást elemeznek, amelyek közül csak 13 esetben cél a hulladék minimalizálása, a többi esetben az üzemi körülmények más célt írnak elő.

Azért, hogy az eljárás széleskörben felhasználható legyen, az (1—3) feladathoz az anyagleszabás általánosítható körülményeit tükröző újabb feltételeket csatolunk. Az új feltételek megadásánál két szempontot tartunk szem előtt: egyrészt azt, hogy a feltételek tükrözzék vissza a *technológiai, műszaki, szervezési körülményeket*, másrészt tegyék lehetővé a *leszabás vezérlését* (a feltételek dinamizálhatók legyenek). Ez utóbbit a feltételekben szereplő *vezérlő paraméterekkel* érhetjük el. A vezérlő paramétereket az ismételt számítógépes futtatások során addig kell módosítani, amíg a gyakorlatban is alkalmazható szabásmintákat nyerünk. Így iterációs lépések után a szabásminták olyan követelményeknek is eleget tesznek, amelyek nem, vagy csak nehezen modellezhetők. Ezzel megszüntethető a leszabás determinisztikussága, a gyártásprogramozó (termelésirányító) döntésévé válik az, hogy melyik szabásminta sorozatot fogadja el, tartja megvalósíthatónak. A leszabást irányító vállalati szakemberek olyan meghatározó — gyakran szubjektív és időben is sűrűn változó — körülményeket is ismernek, amelyeket matematikai formalizmussal nehéz jól leírni.

Az iterációk során a kívánt irányba terelhető a szabásterv, mivel a GGM „stabil" abban az értelemben, hogy a vezérlő paraméterek kis változtatása esetén az optimális célfüggvényérték is kicsit változik (a leszabás nagyméretű lineáris programozási feladata miatt). A heurisztikus szekvenciális eljárások „mohó" eljárások, amelyek a vezérlő paraméterek egyre jobb beállításával jelentősen javíthatók. HAESSLER elnevezése erre az iterációra: „multiple-pass heuristic procedure".

A következő feltételeket vettük figyelembe, kevés és egyszerűen áttekinthető feltétellel a leszabás lényeges jellemzőinek kifejezésére törekedtünk:

$$R_l \leq \sum_{i=1}^{m} x_i \leq R_u \qquad R_l \text{ és } R_u \text{ pozitív egész.} \tag{4}$$

A (4) feltétel mint technológiai feltétel a *vágófejek korlátozott számá*ból adódik akkor, amikor a vágófejek egyszerre vágnak [15]. Az R-eket paraméterként változtatva a szabáskép-sorozat vezérelhető.

$$T_l \leq \sum_{i=1}^{m} \delta(x_i) \leq T_u, \qquad T_l \text{ és } T_u \text{ pozitív egész,} \tag{5}$$

ahol

$$\delta(x_i) = \begin{cases} 1 & \text{ha } x_i > 0, \\ 0 & \text{egyébként.} \end{cases}$$

Az (5) feltételt a *leszabás szervezési feltételé*nek nevezzük. Az üzemi gyakorlatban a legtöbbször azzal a klasszikus esettel találkozunk, amikor egyidőben csak egyféle idomot szabnak ki addig, amíg az igényelt darabszámot el nem érik (azaz $T_l = T_u = 1$). Ekkor a keletkező hulladék nagy, viszont az alapanyagok bekészítése egyszerű (mivel egyféle alapanyag kell), a vágófejek beállítása, az idomok levágása és válogatása, a levágott idomok nyilvántartása, az idomok mozgatása, tárolása könnyen megszervezhető. Lehetőség van az esetleges selejt rögtöni pótlására az alapanyagoknál és az idomoknál is. Amikor a T_u vezérlő paramétert növeljük, akkor jelentősen csökken a hulladék, viszont a vágófejeket sűrűbben kell állítgatni, az idomok válogatása és az aktuális darabszám nyilvántartása, az idomok tárolása, a selejt pótlása is jóval nehezebben szervezhető meg. Az (5) feltétel technológiai jellegű is lehet, amikor a munkahelyek és a tárolóhelyek száma a szabászgép mellett korlátozott, egyféle idomot csak egy tárolóhelyre lehet gyűjteni, általában egy munkás csak egyféle idomot szed le, ellenőriz és tárol (lásd a húzott síküveg szabását [26]).

$$0 \leq l_i \leq x_i \leq u_i \qquad l_i, x_i \text{ és } u_i \text{ egész } (i = 1, 2, \ldots, m). \tag{6}$$

A (6) *egyedi korlátok* mint vezérlő paraméterek működnek ($u_i \leq [L/a_i]$). HAESSLER szerint a GGM-nél a szabásminták száma csökkenthető (ezáltal a vágófejek átállításának idő- és költségigénye is) az egyedi korlátok ügyes megválasztásával [17]. A szekvenciális eljárás javítható, ha a magas fajlagos célfüggvény-együtthatóval rendelkező idomok egyedi korlátait csökkentjük, illetve az idomok túlszabása is vezérelhető (túlszabásnak nevezzük azt, ha a kért idomból a kerekítések miatt a darabszámnál többet vágunk le). Ha $l_i = 0$ és $u_i = 1$, akkor $0-1$ programozási feladat (bináris hátizsákfeladat) áll elő.

Az újabb feltételek csatolásával az $(1-3)$ hátizsákfeladatból megkapjuk az alábbi nemlineáris többfeltételes feladatot:

$$0 \leq l_i \leq x_i \leq u_i \qquad l_i, x_i \text{ és } u_i \text{ egészek } (i = 1, 2, \ldots, m)$$

$$\sum_{i=1}^{m} a_i x_i \leq L \tag{7}$$

$$R_l \leq \sum_{i=1}^{m} x_i \leq R_u \qquad R_l \text{ és } R_u \text{ pozitív egész}$$

$$T_l \leq \sum_{i=1}^{m} \delta(x_i) \leq T_u \qquad T_l \text{ és } T_u \text{ pozitív egész}$$

$$\sum_{i=1}^{m} c_i x_i \to \max.$$

A (7) feladatban csak egyféle alapanyag szerepel. Ha többfajta, eltérő méretű alapanyagot veszünk figyelembe és alapanyagfajtánként változtatjuk a vezérlő paramétereket, valamint az alapanyagok fajlagos egységára is eltérő, akkor kapjuk a következő általános feladatot (illetve feladat-családot):

$$z = \max_j \left\{ \frac{\sum_{i=1}^{m} c_i x_{ji}}{C_j} \middle| \; 0 \leq l_{ji} \leq x_{ji} \leq u_{ji}; \quad l_{ji}, x_{ji}, u_{ji} \text{ egészek } (i = 1,2, \ldots, m); \right.$$

$$\sum_{i=1}^{m} a_i x_{ji} \leq L_j; \; R_{jl} \leq \sum_{i=1}^{m} x_{ji} \leq R_{ju}; \tag{8}$$

$$\left. T_{jl} \leq \sum_{i=1}^{m} \delta(x_{ji}) \leq T_{ju}; \; R_{jl}, R_{ju}, T_{jl}, T_{ju} \text{ pozitív egészek } (j = 1, 2, \ldots, p) \right\},$$

ahol $C_j > 0$ az L_j hosszú alapanyag ára.

A (8) feladat a szabásmintákkal szembeni elvárásokat modellezi. Ez a feladat nem egy konkrét üzem feltételeit adja vissza, hanem az üzemi technológiai-szervezési körülmények absztrakciója. Így lehetőség van olyan programrendszer elkészítésére, amely széles körben felhasználható. Nem kell egyedi modelleket szerkeszteni, hanem a helyi üzemi körülmények figyelembe vehetők a paraméterek inputként való megadásával. A dolgozat lezárásakor a fentieken alapuló programokat már több üzem megvette és alkalmazza.

A GGM-nél a paraméterek előre rögzítettek, míg a heurisztikus szekvenciális eljárásoknál lehetőség van arra is, hogy az egyes szabásminták elfogadása, azaz a (8) feladat megoldása után a paramétereket módosítsuk és a következő szabásmintát már az új értékek, az új (8) feladat szerint keressük meg.

A szakirodalomban hátizsákfeladatnak az (1—3) feladatot nevezik, amely az NP-teljes feladatok családjába tartozik. A hátizsákfeladat hatékony megoldásával több munka foglalkozik, lásd Salkin és De Kluyver összefoglalóját [31], illetve a következő alapvető publikációkat [2], [13], [14], [20], [29], magyar nyelven [3], [6], [23], [24]. Az eljárások döntő része a (6) feltételt is figyelembe veszi (egészértékű feladat visszavezethető binárisra), amely dinamikus programozásnál és a korlátozás és szétválasztás (branch and bound) elvének alkalmazásánál előnyös.

Az anyagleszabással foglalkozó irodalomban Gilmore és Gomory [11] felveti a (7) feladatot (ahol $j = 1$ és $l_i = 0$, $R_l = 0$, $T_l = 0$), amelynek megoldására dinamikus programozáson alapuló eljárást adnak (hátizsákfüggvényt kell realizálni). Marconi [28] az (1—4) feladatnál szintén dinamikus programozást alkalmazott. KUUTTI és VOUTILAINEN [25] a (2), (3), (6) feladatot ismételten megoldva, a (6) feltétel egyedi korlátait addig módosítja, amíg a (4) feltételnek is eleget tevő megoldást kap. Ők MARTELLO és TOTH [29] algoritmusát használják fel. Az egydimenziós leszabás feltételeiről JOHNSTON [21] nyújt jó összefoglalót.

A fejezetben bemutatott technológiai-vezérlő paramétereken alapuló modellezéssel, a (7) feladat dinamizálhatóságával, alsó korlátos változatával, valamint a többfajta alapanyagot és fajlagos költséget figyelembe vevő (8) feladattal az anyagleszabás szakirodalmában nem találkoztam.

3. A feladat megoldására kidolgozott eljárás

A (8) feladatot megoldó eljárás készítésekor a diszkrét programozás hátizsákfeladatokra vonatkozó közismert eredményeiből indultunk ki (lásd magyar nyelven [6], [23]). A (8) feladat megoldását a (7) részfeladat alapanyagonkénti ($j = 1, 2, \ldots, p$) megoldására vezetjük vissza. Lexikografikus rendezésen alapuló implicit leszámlálást (fabejárást) használunk: felsőkorlátok kiszámításával becsüljük azt meg, hogy az egyes részfeladatokat, illetve a részfeladatok lehetséges megoldásainak részhalmazait elhagyhatjuk-e. A feladat NP-teljes, a leszámlálás sebessége legrosszabb esetben exponenciálisan függ a változók számától, ezért az eljárást a számítógépes futtatások során heurisztikus meggondolásokat felhasználva finomítottuk.

A megoldandó j-edik részfeladat:

$$
\begin{gathered}
0 \le l_{ji} \le x_{ji} \le u_{ji} \qquad l_{ji}, x_{ji}, u_{ji} \text{ egészek } (i = 1, 2, \ldots, m) \\
\sum_{i=1}^{m} a_i\, x_{ji} \le L_j \\
R_{jl} \le \sum_{i=1}^{m} x_{ji} \le R_{ju} \\
T_{jl} \le \sum_{i=1}^{m} \delta(x_{ji}) \le T_{ju} \\
\sum_{i=1}^{m} c_i x_{ji} \to \max,
\end{gathered}
\tag{9}
$$

ahol R_{jl}, R_{ju}, T_{jl}, T_{ju} pozitív egészek, $a_i > 0$ és $c_i > 0$ $(i = 1, 2, \ldots, m)$. Képezzük a (9) feladatból az alábbi redukciós feladatot:

$$
\begin{gathered}
0 \le x_{ji} \le u_{ji} - l_{ji} \quad (i = 1, 2, \ldots, m) \\
\sum_{i=1}^{m} a_i x_{ji} \le L_j - \sum_{i=1}^{m} l_{ji}\, a_i \\
\sum_{i=1}^{m} x_{ji} \le R_{ju} - \sum_{i=1}^{m} l_{ji} \\
\sum_{i=1}^{m} \delta(x_{ji}) \le T_{ju} - \sum_{i=1}^{m} \delta(l_{ji})
\end{gathered}
\tag{10}
$$

$$\sum_{i=1}^{m} x_i\, x_{ji} \to \max.$$

Tegyük fel, hogy a (10) feladat megengedett megoldásainak halmaza nem üres és optimális megoldása $(\bar{x}_{j1}, \bar{x}_{j2}, \ldots, \bar{x}_{jm})$.
Ekkor nyilvánvalóan teljesül az alábbi lemma:

1. *lemma.* A (9) feladat célfüggvényének a $\sum_{i=1}^{m} c_i(\bar{x}_{ji} + l_{ji})$ mennyiség egy felső korlátja. Ha

$$R_{jl} - \sum_{i=1}^{m} (l_{ji} + \bar{x}_{ji}) \leq 0 \text{ és } T_{jl} - \sum_{i=1}^{m} [\, \delta(l_{ji}) + \delta(\bar{x}_{ji})] \leq 0,$$

akkor a $(\bar{x}_{j1} + l_{j1}, \bar{x}_{j2} + l_{j2}, \ldots, \bar{x}_{jm} + l_{jm})$ vektor a (9) feladat optimális megoldása.

A fentiek miatt a továbbiakban a redukált (10) feladattal foglalkozunk, amelynek felső korlátait alkalmazzuk a (9) feladat célfüggvényének becsléséhez. Megjegyezzük, hogy egyes esetekben a (9) feladatra ez a becslés durva (pl. ha $R_{jl} - \sum_{i=1}^{m} l_{ji} > 0$ és $T_{jl} - \sum_{i=1}^{m} \delta(l_{ji}) > 0$), azonban előnye, hogy gyorsan és egyszerűen kiszámolható.

Rendezzük át az alapanyagokat és az idomokat:

$$L_1/C_1 \geq L_2/C_2 \geq \ldots \geq L_p/C_p \text{ és} \qquad (11)$$
$$c_1/a_1 \geq c_2/a_2 \geq \ldots \geq c_m/a_m.$$

Az egyszerűbb írásmód kedvéért a (10) feladatot az alábbi formába írjuk át, ahol a korábbi jelöléseket használjuk (de már redukált értékkel):

$$\begin{gathered} 0 \leq x_{ji} \leq u_{ji} \qquad x_{ji},\, u_{ji} \text{ egészek } (i = 1, 2, \ldots, m) \\ \sum_{i=1}^{m} a_i x_{ji} \leq L_j \\ \sum_{i=1}^{m} x_{ji} \leq R_{ju} \qquad (12) \\ \sum_{i=1}^{m} \delta(x_{ji}) \leq T_{ju} \\ \sum_{i=1}^{m} c_i x_{ji} \to \max. \end{gathered}$$

A (8) feladat optimális megoldását úgy kapjuk meg, hogy először $j = 1$, majd $j = 2, \ldots, j = p$ esetben megoldjuk egyenként a (12) részfeladatot. Az eljárás szekvenciális felépítése miatt két becslést kell készítenünk. A j-edik részfeladat célfüggvényértékére adunk egy felső korlátot, amellyel azt becsüljük meg, hogy a (8) feladatra az eddigi legjobb megoldásnál ez a részfeladat adhat-e jobbat. Ha igen, akkor a j-edik feladat egyes részhalmazait teszteljük szintén felső korlátokat megadva. Az egyes részhalmazok vizsgálatánál Gil-

MORE és GOMORY lexikografikus rendezésen alapuló eljárását [9] fejlesztettük tovább, amely a folytonos feladat megoldását használja fel. (Folytonosnak nevezzük a feladatot, ha az egészértékűségtől eltekintünk.)

Előbb a legszükségesebb fogalmakat és jelöléseket adjuk meg. Tekintsük az $x = (x_1, x_2, \ldots, x_m)$ és az $y = (y_1, y_2, \ldots, y_m)$ nemnegatív vektorokat.

Definíció: Az x vektor *lexikografikus rendezésben követi* az y vektort (vagy nagyobb nála), ha van olyan q, amely $1 \leq q \leq m$ és $x_1 = y_1, x_2 = y_2, \ldots, x_{q-1} = y_{q-1}$ és $x_q > y_q$.

Definíció: Az y vektor *q-szerint lexikografikusan kisebb* az x vektornál $(1 \leq q \leq m)$, ha $y_1 = x_1, y_2 = x_2, \ldots, y_{q-1} = x_{q-1}$ és $y_q < x_q$.

Jelölje a v valós szám *egész részét* $[\,]^-$, azaz a v-nél nem nagyobb legnagyobb egész számot.

Készítsük el a (12) feladat megengedett megoldásai közül a lexikografikus rendezés szerinti legnagyobbat:

$$\bar{x}_{j1} = \min \{[L_j/a_1]^-, u_{j1}, R_{ju}\}$$

$$\bar{x}_{jn} = \begin{cases} \min\left\{\left[\left(L_j - \sum_{i=1}^{n-1} a_i \bar{x}_{ji}\right)/a_n\right], u_{jn}, R_{ju} - \sum_{i=1}^{n-1} \bar{x}_{ji}\right\} & \text{ha } \sum_{i=1}^{n-1} \delta(\bar{x}_{ji}) < T_{ju}, \\ 0 & \text{egyébként.} \end{cases} \tag{13}$$

$$(n = 2, 3, \ldots, m).$$

Legyen $r = \max \{i \mid \bar{x}_{jn} = u_{jn}\ (n = 1, 2, \ldots, i)\}$.

A következőkben a 0—1 programozási feladatok körében közismert két lemma általánosítását adjuk meg (lásd pl. [23]).

2. *lemma.* Ha $\sum_{i=1}^{r} a_i \bar{x}_{ji} = L_j$,

akkor a (12) feladat optimális megoldása

$$x_{j1} = \bar{x}_{j1}, x_{j2} = \bar{x}_{j2}, \ldots, x_{jr} = \bar{x}_{jr} \text{ és } x_{ji} = 0\ (i = r+1, \ldots, m).$$

Jelölje P_j a (12) feladat optimális célfüggvényértékét.

3. *lemma.* Ha $\sum_{i=1}^{r} a_i \bar{x}_{ji} < L_j$, akkor

$$P_j \leq \sum_{i=1}^{r} c_i \bar{x}_{ji} + \frac{L_j - \sum_{i=1}^{r} a_i u_{ji}}{c_{r+1}} c_{r+1}.$$

Mindkét lemma bizonyítása triviális.

Vezessük be a célfüggvényérték korlátjának becslésére a következő jelölést, ahol az első r változó értékét lekötöttük (legyen $a_{m+1} = 1$ és $c_{m+1} = 0$):

$$M_j(x_{j1}, x_{j2}, \ldots, x_{jr}) = \sum_{i=1}^{r} c_i x_{ji} + \frac{L_j - \sum_{i=1}^{r} a_i x_{ji}}{a_{r+1}} c_{r+1}.$$

A (8) feladat optimális megoldását úgy kapjuk meg, hogy sorrendben $j = 1, 2, \ldots, p$-re megoldjuk a (12) feladatot. Jelölje Z_{opt} az eddigi legjobb hányadosértéket a (8) feladatra (célfüggvényérték osztva a megfelelő alapanyag árával).

A (13) képzési szabállyal készítsük el az x_{ji} $(i = 1, 2, \ldots, m)$ mennyiségeket $(j > 1)$. Legyen $r = \max \{i \mid x_{jn} = u_{jn}(n = 1, 2, \ldots, i)\}$.

3.1. *következmény.* Ha $Z_{\text{opt}} \geq \left[M_j(x_{j1}, \ldots, x_{jr}) + \sum_{i=1}^{m} c_i\, l_{ji}\right] \Big/ C_j$, akkor a j-edik részfeladat a (8) feladatra nem szolgáltat jobb megoldást a meglevőnél, tehát a j-edik részfeladatot el lehet hagyni.

Az előbbiekben a (12) feladat célfüggvényére adtunk egy felső korlátot, a továbbiakban a (12) feladat megengedett megoldásainak részhalmazait vizsgáljuk, tartalmazhatják-e az optimális megoldást.

A (13) szabállyal képzett lexikografikusan legnagyobb vektorból indulunk ki. Tegyük fel, hogy eljutottunk az $x_j = (x_{j1}, x_{j2}, \ldots, x_{jm})$ megengedett megoldáshoz. Legyen $q = \max \{i \mid x_{ji} > 0 \ (i = 1, 2, \ldots, m)\}$ és a (12) feladatra megtalált eddigi legjobb megoldás célfüggvényértéke: Z_j.

4. *lemma.* Ha $Z_j \geq M_j(x_{j1}, \ldots, x_{jq} - 1)$,
akkor az x_j vektornál q-szerint lexikografikusan kisebb vektorok a (12) feladatra nem adnak jobb megoldást az eddigieknél.

Bizonyítás: Azt kell belátni, hogy az x_j vektornál q-szerint lexikografikusan kisebb $\bar{x}_{j1} = x_{j1}, \ldots, \bar{x}_{jq-1} = x_{jq-1}, \bar{x}_{jq} = x_{jq-s}$, $\bar{x}_{ji} \geq 0$ (ahol $1 \leq s \leq x_{jq}$ és $i = q + 1, \ldots, m$) megengedett megoldásra

$$M_j(x_{j1}, \ldots, x_{jq} - 1) \geq \sum_{i=1}^{m} c_i \bar{x}_{ji}.$$

a) Először azt látjuk be, hogy

$$M_j(x_{j1}, \ldots, x_{jq} - 1) \geq M_j(x_{j1}, \ldots, x_{jq} - s).$$

Tegyük fel, hogy az egyenlőtlenség nem teljesül, azaz kifejtve:

$$\sum_{i=1}^{q-1} c_i\, x_{ji} + c_q(x_{jq} - 1) + \frac{L_j - \sum_{i=1}^{q-1} a_i\, x_{ji} - a_q\,(x_{jq} - 1)}{a_{q+1}}\, c_{q+1} <$$

$$< \sum_{i=1}^{q-1} c_i\, x_{ji} + c_q(x_{jq} - s) + \frac{L_j - \sum_{i=1}^{q-1} a_i x_{ji} - a_q(x_{jq} - s)}{a_{q+1}}\, c_{q+1}.$$

Egyszerűsítve

$$(s - 1)\, c_q a_{q+1} - (x_{jq} - 1) a_q\, c_{q+1} < -\,(x_{jq} - s)\, a_q\, c_{q+1}$$

$$\frac{c_q}{a_q} < \frac{c_{q+1}}{a_{q+1}},$$

ami ellentmond a (11) rendezésnek.

b) Konstruáljuk meg a (12) feladat alapján az alábbi egészértékű feladatot:

$$0 \leq \bar{x}_{jq} \leq x_{jq} - s, \quad 0 \leq \bar{x}_{ji} \leq u_{ji} \; (i = q+1, \ldots, m)$$

$$\sum_{i=q}^{m} a_i \bar{x}_{ji} \leq L_j - \sum_{i=1}^{q-1} a_i x_{ji} - a_q(x_{jq} - s)$$

$$\sum_{i=q}^{m} \bar{x}_{ji} \leq R_{ju} - \sum_{i=1}^{q-1} x_{ji} - (x_{jq} - s) \tag{14}$$

$$\sum_{i=q}^{m} \delta(\bar{x}_{ji}) \leq T_{ju} - \sum_{i=1}^{q} \delta(x_{ji})$$

$$\sum_{i=q}^{m} c_i \bar{x}_{ji} \to \max.$$

A fenti feladatnak legyen $\bar{x}_{jq} = x_{jq} - s, \bar{x}_{ji} \geq 0 \; (i = q+1, \ldots, m)$ egy megengedett megoldása. Nyilvánvaló, hogy a (12) feladatnak $x_{j1}, \ldots, x_{jq-1}$, $\bar{x}_{jq}, \bar{x}_{ji} (i = q+1, \ldots, m)$ x_j-nél q-szerint kisebb megengedett megoldása. A (14) feladatnak

$$\frac{L_j - \sum_{i=1}^{q-1} a_i x_{ji} - a_q(x_{jq} - s)}{a_{q+1}} \, c_{q+1}$$

egy triviális felső korlátja, ezért

$$\frac{L_j - \sum_{i=1}^{q-1} a_i x_{ji} - a_q(x_{jq} - s)}{a_{q+1}} \, c_{q+1} \geq \sum_{i=q}^{m} c_i \bar{x}_{ji}.$$

Az egyenlőtlenség mindkét oldalához a

$$\sum_{i=1}^{q-1} c_i \, x_{ji} + c_q(x_{jq} - s)$$

mennyiséget adva (ahol $\bar{x}_{j1} = x_{j1}, \ldots, \bar{x}_{jq} = x_{jq} - s$), kapjuk, hogy

$$M_j(x_{j1}, \ldots, x_{jq} - s) \geq \sum_{i=1}^{m} c_i \bar{x}_{ji}.$$

Az egyenlőtlenségből az a) részben bizonyítottakat felhasználva a lemma állítása adódik.

4.1. *következmény.* Ha $Z_{\text{opt}} \geq \left(M_j(x_{j1}, \ldots, x_{jq} - 1) + \sum_{i=1}^{m} c_i \, l_{ji}\right) \Big/ C_j$, akkor az x_j vektornál q-szerint lexikografikusan kisebb vektorok a (8) feladatra nem adnak jobb megoldást az eddiginél.

4.2. *következmény.* Ha $Z_{\text{opt}} \geq \left(M_j(x_{j1}, \ldots, x_{jq} - 1) + \sum_{i=1}^{m} c_i \, l_{ji}\right) \Big/ C_j$ és $r = \max \{i \mid x_{ji} > 0 \; (i = 1, 2, \ldots, q-1)\}$ akkor elég az $x_{j1}, \ldots, x_{jr} - 1$ kötött változók mellett a 4.1. következmény feltételét megvizsgálni.

A fenti eredmények birtokában térjünk vissza a (8) feladathoz, amelynek optimális megoldását (ha létezik) a következő eljárás szolgáltatja (a : = jelölés programozási értelemben értékadást jelöl):

0. *lépés.* (kezdeti értékek megadása)

Legyen $j := 0$ és $Z_{\text{opt}} := 0$.

1. *lépés.* (előzetes tesztelés és redukálás)

Legyen $j := j + 1$, ha $j = p + 1$ akkor STOP.

Legyen $L_j := L_j - \sum_{i=1}^{m} a_i l_{ji}$ és $a' = \min\{a_i \mid l_{ji} < u_{ji}\ (i = 1, 2, \ldots, m)\}$

$$R_{ju} := R_{ju} - \sum_{i=1}^{m} l_{ji};\ T_{ju} := T_{ju} - \sum_{i=1}^{m} \delta(l_{ji})$$

$$u_{ji} := u_{ji} - l_{ji} \quad (i = 1, 2, \ldots, m).$$

Ha $a' < L_j$, $0 < R_{ju}$, $0 < T_{ju}$ és $0 \leq u_{ji} (i = 1, 2, \ldots, m)$
akkor menj a 2. lépésre,
egyébként menj az 1. lépésre.

2. *lépés.* (induló megoldás elkészítése)

Legyen $x_{j1} := \min\{[L_j/a_1]^-, u_{j1}, R_{ju}\}$

$$x_{jn} := \begin{cases} \min\left\{\left[\left(L_j - \sum_{i=1}^{n-1} a_i x_{ji}\right)\Big/ a_n\right]^-, u_{jn}, R_{ju} - \sum_{i=1}^{n-1} x_{ji}\right\} & \text{ha } \sum_{i=1}^{n-1} \delta(x_{ji}) < T_{ju} \\ 0 \text{ egyébként} & \end{cases}$$

$(n = 2, 3, \ldots, m)$.

Legyen $a_{m+1} := 1$ és $c_{m+1} := 0$.

3. *lépés.* (j-edik részfeladat célfüggvényének becslése)

Legyen $q := \max\{i \mid x_{jn} = u_{jn}\ (n = 1, 2, \ldots, i)\}$.

Ha $Z_{\text{opt}} \geq \left(M_j(x_{j1}, \ldots, x_{jq}) + \sum_{i=1}^{m} c_i l_{ji}\right)\Big/ C_j$,
akkor menj az 1. lépésre,
egyébként menj a 4. lépésre.

4. *lépés.* (egy megengedett megoldás vizsgálata)

Ha $Z_{\text{opt}} < \left(\sum_{i=1}^{m} c_i(x_{ji} + l_{ji})\right)\Big/ C_j$ és $R_{jl} \leq \sum_{i=1}^{m} (x_{ji} + l_{ji})$, $T_{jl} \leq \sum_{i=1}^{m} \delta(x_{ji} + l_{ji})$

akkor legyen $Z_{\text{opt}} := \left(\sum_{i=1}^{m} c_i (x_{ji} + l_{ji})\right)\Big/ C_j$ és tároljuk az

$(x_{j1} + l_{i1}, x_{j2} + l_{j2}, \ldots, x_{jm} + l_{jm})$ vektort,

egyébként menj az 5. lépésre.

5. *lépés.* (megengedett megoldások részhalmazának tesztelése)

5.1. legyen $q := \max \{i \mid x_{ji} > 0 \ (i = 1, 2, \ldots, m)\}$.

Ha $\sum_{i=1}^{m} \delta(x_{ji}) = T_{ju}$ vagy $q = m$ akkor legyen $x_{jq} = 0$.

5.2. Legyen $q := \max \{i \mid x_{ji} > 0 \ (i = 1, 2, \ldots, m)\}$.

Ha $Z_{\text{opt}} < \left(M_j(x_{j1}, \ldots, x_{jq} - 1) + \sum_{i=1}^{m} c_i l_{ji}\right) \Big/ C_j$,

akkor menj a 6. lépésre,
egyébként legyen $x_{jq} := 0$.

5.3. Ha $x_{ji} = 0 \ (i = 1, 2, \ldots, m)$ akkor menj az 1. lépésre,
egyébként menj az 5.2. lépésre.

6. lépés. (q-szerint lexikografikusan kisebb vektor előállítása)

$$\text{Legyen } x_{jq+1} = \min \left\{ \left[\left(L_j - \sum_{i=1}^{q} a_i x_{ji} \right) \Big/ a_{q+1} \right]^{-}, u_{jq+1}, R_{ju} - \sum_{i=1}^{q} x_{ji} \right\}$$

$$x_{jn} := \begin{cases} \min \left\{ \left[\left(L_j - \sum_{i=1}^{n-1} a_i x_{ji} \right) \Big/ a_n \right]^{-}, u_{jn}, R_{ju} - \sum_{i=1}^{n-1} x_{ji} \right\} & \text{ha } \sum_{i=1}^{n-1} \delta(x_{ji}) < T_{ju} \\ 0 \text{ egyébként.} & \end{cases}$$

$$(n = q + 2, \ldots, m).$$

Menj a 4. lépésre.

Ha az eljárás végeztével $Z_{\text{opt}} = 0$, akkor a (8) feladatnak nincs lehetséges megoldása, egyébként előállítottuk az optimális megoldást.

5. *állítás.* A fenti eljárás véges sok lépésben végetér és a (8) feladatra optimális megoldást szolgáltat.

Bizonyítás: a) A lexikografikus rendezés szerinti legnagyobb vektorból kiindulva mindig lexikografikusan kisebb vektorokat vizsgálunk az 5.2. lépésben, ezért egy vektor csak egyszer fordulhat elő. Mivel a lehetséges megoldások száma is véges, ezért az eljárás véges sok lépésben végetér.

b) A (8) feladat megengedett megoldásainak részhalmazait a 3. lépésben a 3.1. következményt, az 5.2. lépésben a 4.1. és 4.2. következményt felhasználva hagyjuk el, így az optimális megoldást nem tartalmazhatják az elhagyott részhalmazok.

Az eljárás a (8) feladatot $0, 1, \ldots, R_{jl}$ és $0, 1, \ldots, T_{jl}$ $(j = 1, 2, \ldots, p)$ alsó korlátokra is megvizsgálja, azonban csak a megfelelő alsó korlátoknak eleget tevő vektorokat fogadja el a 4. lépésben.

A fenti alapeljárás gyorsítható az alábbiak felhasználásával:

(i) Legyen $\text{MIN}_i = \min \{a_n \mid i \le n \le m\}$ és $\text{MAX}_i = \max \{a_n \mid i \le n \le m\}$.

Az 5.2. lépésben a felsőkorlát becslésénél a

$$\text{MIN}_{q+1} \le L_j - \sum_{i=1}^{q} a_i x_{ji}$$

$$\text{és a } Z_{\text{opt}} \le \left(\sum_{i=1}^{q} c_i x_{ji} + \left(R_{ju} - \sum_{i=1}^{q} x_{ji} \right) \text{MAX}_{q+1} \frac{c_{q+1}}{a_{q+1}} \right) \Big/ C_j$$

feltételeket is figyelembe vesszük, ekkor a fabejárásnál nem kell lemenni a levélig akkor sem, ha a folytonos megoldásból adódó becslés ezt lehetővé tenné.

(ii) Egyszerű eszközökkel igazolható, ha $a_i \leq a_n$ és $c_i \geq c_n$, valamint $u_{ji}\, a_i + a_n > L_j$, akkor $x_{jn} = 0$. Ezáltal a feladat mérete (az m) csökkenthető a 2. lépésben.

4. Az eljárás heurisztikus módosítása

A 3. fejezetben ismertetett eljárás a (8) feladat optimális megoldását szolgáltatja, azonban a leszabási módszerek többségénél elég egy *jó közelítő megoldás* megkeresése. A GGM-nél egy vektor bázisba vonhatóságának feltétele a megfelelő árnyékköltség pozitivitása, a

$$\sum_{i=1}^{m} c_i x_{ji} - C_j > 0$$

egyenlőtlenség fennállása. Heurisztikus szekvenciális eljárásoknál is előnyösebb az eljárás ,,mohósága'' miatt egy megfelelő közelítő megoldás elkészítése. Mivel a vezérlő paraméterek módosítgatásával a leszabás pillanatnyi üzemi feltételeit kielégítő szabásmintákat akarunk előállítani, ezért a darabolás helyén vagy onnan elérhető távolságban kell a számítógépnek lennie. Napjainkban ez helyben levő személyi, illetve mikroszámítógépekkel oldható meg, amelyeknek teljesítményére tekintettel kell lenni az algoritmusok és a számítógépes programok tervezése során (távadatfeldolgozásra, nagyobb számítógéphez való real-time csatlakozásra egy üzemnek kicsi az esélye).

A fentiek miatt az eljárást heurisztikus ötleteket felhasználva módosítottuk, ezáltal egy elfogadható közelítő megoldás gyorsan megtalálható (futtatási eredményeket az 5. fejezetben ismertetünk).

1. A felső korlátok eredeti becslése durvának bizonyult, ezért a 3. és az 5.2. lépésben $M_j(x_{j1}, \ldots, x_{jq})$ — t módosítottuk:

$$M_j(x_{j1}, \ldots, x_{jq}) = \sum_{i=1}^{q} c_i x_{ji} + \left[\left(L_j - \sum_{i=1}^{q} a_i x_{ji}\right) c_{q+1}/a_{q+1}\right]^{-}.$$

2. A j-edik részfeladat célfüggvényértékének becsléséhez a lexikografikusan legnagyobb vektort használjuk fel, azaz a 3. lépésben: $q := \max\{i \mid x_{ji} > 0\ (i = 1, 2, \ldots, m)\}$.
3. Egy *elfogadási korlátot* ($\varepsilon > 0$) állítunk fel (amely lehet bizonyos fajlagos mutató, pl. a hulladék %-os aránya, a GGM-nél az árnyékköltség nagysága stb.). A 4. lépésben az első olyan megengedett megoldást elfogadjuk (és az eljárásból kiugrunk), amely a megadott elfogadási korlátnál nem rosszabb.

5. A számítógépes juttatások tapasztalatai

Az alapeljárást a 3. fejezet végén közölt (i) gyorsító becslésekkel (MIN_i és MAX_i) egészítettük ki, a heurisztikus módosítások közül 1. és 2., illetve esetenként a 3. változtatást vettük figyelembe. Az algoritmusra Commodore-64 mikroszámítógépen készült BASIC nyelvű program, amelyet gépi kódra fordítottunk le.

1. táblázat

Vezérlő paraméterek értékei

	R_{ju}	T_{ju}	u_{ji}	ε
a)	∞	∞	∞	—
b)	10	∞	∞	—
c)	∞	3	∞	—
d)	∞	∞	3	—
e)	10	3	3	—
f)	∞	∞	∞	$0{,}001\,L_j$

2. táblázat

Az átlagos futásidő másodpercben

	$m=50$			$m=100$			$m=200$		
	$p=5$	$p=20$	$p=50$	$p=5$	$p=20$	$p=50$	$p=5$	$p=20$	$p=50$
a)	9,57	17,31	32,59	14,45	32,98	76,98	32,55	45,30	93,74
b)	5,61	10,34	22,65	6,51	17,31	39,29	8,01	27,06	73,62
c)	6,74	12,61	28,56	14,55	29,08	48,27	14,87	31,18	79,34
d)	14,24	33,34	87,22	20,96	49,01	83,11	62,64	87,49	124,92
e)	27,59	58,70	78,98	53,19	82,79	109,36	60,45	130,45	164,04
f)	3,45	8,03	9,21	3,69	4,17	7,71	3,36	5,05	15,90

A (8) feladatot a 2. táblázatban található eltérő feladatméretek (p és m) mellett oldottuk meg. A paraméterek és változók alsó korlátait zérusnak vettük, az együtthatók az alábbi korlátok között levő *valós számok* voltak, amelyeket a számítógép egyenletes eloszlású véletlenszámgenerátorával állítottunk elő:

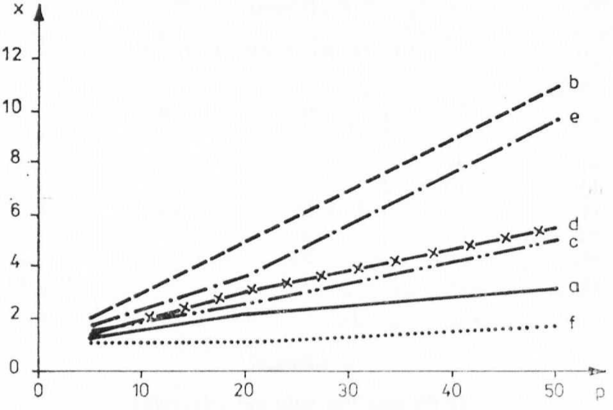
$$\begin{aligned} 1000 \leq L_j &\leq 3000 \\ L_j - 200 \leq C_j &\leq L_j + 200 \\ 10 \leq a_i &\leq 400 \\ a_i \leq c_i &\leq 5a_i . \end{aligned}$$

Feladatonként az 1. táblázatban megadott 6 esetet vettük figyelembe, ahol az (f) esetben ε küszöbszám a 0,1%-os hulladékot (felületveszteséget) jelenti. Feladatméretenként és esetenként 10 feladatot generáltunk, az átlagos futásidők a 2. táblázatban találhatók (az időket másodpercben adtuk meg). A szórások jelentősek voltak, ezért az átlagok csak mérvadó becslések.

A 2. táblázatban közölt futásidőkhöz bizonyos támpontot nyújt a (11) rendezéshez szükséges idő (ahol a kapcsolódó tömböket is rendeztük), amely $m = 50$ és $p = 50$ esetben 30—33 sec, $m = 100$ és $p = 50$ esetben 60—65 sec míg $m = 200$ és $p = 50$ esetben 180—190 sec buborékrendező algoritmussal.

Az 1. ábra szemlélteti azt, hogy a vezérlő paraméterek eltérő értékei esetén a $j = 1$-hez szükséges időnek hányszorosa kell a feladat teljes megoldásához.

Az (f) esetben, ha $L_j - \sum_{i=1}^{m} a_i x_{ji} \leq 0{,}001\, L_j$, akkor az eljárás befejeződik. Egy alkalmas küszöbszámmal a feladatnak egy közelítő megoldása, azaz egy elfogadható szabásminta gyorsan előállítható. Mivel az üzemi gyakorlatban



1. ábra. Az 1. alapanyag idejének átlagosan hányszorosa szükséges a teljes feladat megoldásához ($m = 100$)

hulladékmentes leszabás ritkán fordul elő, ezért egy megfelelő küszöbszám az esetek többségében megadható.

Az algoritmus időigénye legnagyobb az (e) esetben, amikor minden feltétel szoros, és a (d) esetben, ahol az egyedi korlátok kis értékűek. Ha az R_{ju} vagy T_{ju} paramétert vesszük szűkösnek [(b) és (c) eset], akkor a korlát nélküli (a) esetnél csak kevéssel kapunk jobb futásidőt. Szembetűnő, hogy az (f) esetben (ahol $\varepsilon = 0{,}001$) mennyire kicsi az időszükséglet, az (e) esethez képest egy nagyságrenddel jobb. Az 1. ábrából látható, hogy az (f) esetben általában már az első alapanyagra találunk elfogadható megoldást, nem kell minden alapanyagot végignézni.

Megjegyezzük, hogy a 3. fejezet végén megadott (i) becslések (MIN_i és MAX_i) a (b) esetben előnyösek, mivel használatukkal egy nagyságrenddel javult a leszámlálás időszükséglete.

Az algoritmus programját három programrendszer tartalmazza. Egydimenziós esetben egyrészt a GGM-ben, másrészt egy heurisztikus szekvenciális eljárásban szerepel. A harmadik programrendszer a kétdimenziós két-ütemű guillotine-vágásra készült, amely heurisztikus eljáráson alapszik. Ebben az esetben nem a (8), hanem a (7) feladatot kell minden csíkszélességre és alaptáblaméretre megoldani [ha minden idom és alaptábla elforgatható, akkor a (7) feladatot maximálisan $2p$ $(2m + 1)$-szer kell megoldani]. Az alábbi vezérlő paramétereket vettük még figyelembe:

- csíkban az idomok maximális számát,
- csíkban az idomféleségek maximális számát,
- a csíkok maximális számát egy alaptáblán,
- az eltérő csíkok maximális számát egy alaptáblán.

Köszönetnyilvánítás: Ezúton mondok köszönetet a Békés megyei Tanács Tudományos Koordinációs Szakbizottságának, elsősorban *Dr. Köteles Lajos* titkárnak azért, hogy ösztöndíjukkal lehetővé tették a témakörben való elmélyülést, a fenti eredmények kidolgozását.

(Beérkezett: 1986. február 15-én.)

IRODALOM

1. Adamowicz, M.—Albano, A.: A solution of the rectangular cuttingstock problem. *IEEE Transactions on Sys. Man. Cyb.* 6 (1976), 302—310.
2. Akinc, U.: An algorithm for the knapsack problem. *IIE Transactions* 15 (1983), 31—36.
3. Biró M.: A bináris hátizsák feladat. *Alkalmazott Mat. Lapok* 9 (1983) 1—2., 113—136.
4. Dowsland, W. B.: Two and three dimensional packing problems and solution methods. *NZ Oper. Res.* 13 (1985), 1—18.
5. Dyckhoff, H.—Kruse, H. J.—Abel, D.—Gal, T.: Trim loss and related problems. *Omega* 13 (1985), 59—72.
6. Forgó F.: *Nemkonvex és diszkrét programozás.* Közgazdasági és Jogi Kk., Budapest, 1978.
7. Gilmore, P. C.: Cutting stock, linear programming, knapsacking, dynamic programming and integer programming, some interconnections. *Annals of Discrete Mathematics 4* (1979), 217—235.
8. Gilmore, P. C.—Gomory, R. E.: A linear programming approach to the cutting stock problem. *Operations Research* 9 (1961), 849—859.
9. Gilmore, P. C.—Gomory, R. E.: A linear programming approach to the cutting stock problem. — Part II. *Operations Reseach* 11 (1963), 863—888.
10. Gilmore, P. C.—Gomory, R. E.: Multistage cutting stock problems of two and more dimensions. *Operations Research* 13 (1965), 94—120.
11. Gilmore, P. C.—Gomory, R. E.: The theory and computation of knapsack functions. *Operations Research* 14 (1966), 1045—1074.
12. Golden, B. L.: Approaches to the cutting stock problem. *AIIE Transactions* 8 (1976), 265—274.
13. Greenberg, H.—Hegerich, R. L.: A branch algorithm for the knapsack problem. *Management Science* 16 (1970), 327—332.
14. Greenberg, H.—Feldman, I.: Better step-off algorithm for the knapsack problem. *Discr. Appl. Math.* 2 (1980), 21—25.
15. Haessler, R. W.: An heuristic programming solution to a non-linear cutting stock problem. *Management Science* 17 (1971), B793-B802.
16. Haessler, R. W.: Controlling cutting pattern changes in one-dimensional trim problems. *Operations Research* 23 (1975), 483—493.
17. Haessler, R. W.: A note on computational modifications to the Gilmore-Gomory cutting stock algorithm. *Operations Research* 28 (1980), 1001—1004.
18. Hinxman, A. I.: A two-dimensional trim-loss problem with sequencing constraints. *Advance papers of IJCAI-77, MIT* (1977), 859—864.
19. Hinxman, A. I.: The trim-loss and assortment problems: a survey. *European Journal of Opl. Res.* 5 (1980), 8—18.
20. Ingargiola, G. P.—Korsh, J. F.: A general algorithm for one-dimensional knapsack problems. *Operations Research* 25 (1977), 752—759.
21. Johnston, R. E.: Bounds for the one dimensional cutting stock problem. *Working paper*, Footscray Ins. of Tech., Victoria, Australia (dec. 1980).
22. Kalmár J.—Lengyel I.: A raktárkészlet figyelembevétele a Gilmore-Gomory módszernél. *Szigma* 17 (1984), 167—183.
23. Kovács L. B.: *A diszkrét programozás kombinatorikus módszerei.* Bolyai J. Matematikai Társulat jegyzete (1969).
24. Krekó B.: *Optimumszámítás.* Közgazdasági és Jogi Kk., Budapest (1972).
25. Kuutti, M.—Voutilainen, R.: A discrete optimization approach to the pattern analysis of the cutting stock problem. Paper presented at EURO III. Amsterdam (1979).
26. Lengyel I.—Kuba A.: Síküveg szabásának optimalizálása. (I). *Szigma* 14 (1981), 169—190.
27. Lengyel I.: Darabolás — korszerű módszerekkel. *Ipargazdaság* (1984. nov.), 19—31.
28. Marconi, R.: Generalisation of a mathematical procedure for the optimal solution of two-dimensional cutting problems. *Technical reports* No. 7., Centro Studi IBM — Pisa (1970).
29. Martello, S.—Tóth P.: Branch and bound algorithms for the solution of the general unidimensional knapsack problem. *Proceedings of the EURO II.* Stockholm (1976).
30. Pegels, C. C.: Heuristic scheduling models for variant of two-dimensional cutting stock problem. *Tappi* 50 (1967), 532—535.

31. Salkin, H. M. — De Kluyver, C. A.: The knapsack problem: a survey. *Naval Research Log. Quart.* 22 (1975), 127—144.
32. Tilanus, C. B. — Gerhardt, C.: An application of cutting stock in the steel industry. in: *Operational Research '75* (ed: K. B. Haley), North-Holland, Amsterdam (1976), 669—675.

A PRACTICAL GENERALIZATION OF THE KNAPSACK PROBLEM FOR CUTTING STOCK

For solving the cutting problem, in addition to reduction to the commonly known linear programming problem, several heuristical procedures have also been worked out. From the methods applied, two approaches, most frequently used in practice, may be underlined:

1. The Gilmore—Gomory method which solves the linear programming problem of cutting with a modified simplex method, so that the vector to be drawn into the base is produced by solving knapsack problems.

2. A heuristical sequential procedure (emptying procedure) is the one which produces a "best" cutting pattern according to some criterion, applying this pattern as many times as possible and then produces the "best" pattern for the remaining basic materials and profiles etc. The "best" cutting pattern is usually constructed from the exact or approximative solution to the knapsack problem.

The paper reviews a novel modelling of the practical circumstances of cutting materials, the problem derived as a generalization of the knapsack problem, and the procedure worked out for the solution of the problem. From among the cutting problems the generalization of the knapsack problem of one-dimensional cutting (paper rolls, bars, tubes, bands, etc.) is discussed in detail, which is the same in the case of both the Gilmore-Gomory method and the heuristical sequential procedure.

ПРАКТИЧЕСКИЕ ОБОБЩЕНИЕ ЗАДАЧИ О РАНЦЕ ДЛЯ ПРОБЛЕМЫ РАСКРОЯ МАТЕРИАЛОВ

Для решения задачи раскроя материалов наряду с общеизвестными методами приведения к задаче линейного программирования разработан также целый ряд эвристических методов. Из применяемых методов автором выделяется два наиболее распространенных метода:

1. Метод Жилмора—Гомори (Gilmore—Gomory), решающий задачу линейного программирования для раскроя с помощью модифицированного симплексного метода так, что вектор, включаемый в базис, определяется из решения задачи о ранце.

2. Эвристический секвенциальный метод, который определяет с некоторой точки зрения „наилучший" раскрой, этот раскрой применяется как можно чаще, потом из оставшегося материала также определяется „наилучший" раскрой и т. д. „Наилучший" раскрой конструируется из точного или приближенного решения задачи о ранце.

В статье рассматривается новая формулировка модели для практического раскроя материалов, обобщение задачи о ранце и метод его решения. Среди задач о раскрое материалов подробно рассматривается одноразмерная задача о раскрое (рулон бумаги, стержки, трубы, ленты и т. д.) с помощью обобщение задачи о ранце, которое применяется в методе Жилмора — Гомори, а также в эвристическом секвенциальном методе.