

A SZABASI FELADAT ÉS AZ EHHEZ KAPCSOLÓDÓ LEFEDÉSI PROBLÉMÁK
MEGOLDÁSA EGZAKT ÉS HEURISZTIKUS MÓDSZEREKKEL

GALAMBOS GÁBOR

Készült a JATE Kibernetikai Laboratóriumban
Dr. Lovász László egyetemi tanár és dr. Csirik János tudományos főmunkatárs irányításával

SZEGED

1981



TARTALOMJEGYZÉK

BEVEZETÉS	1 oldal
1. EGZAKT MEGOLDÁSI MÓDSZEREK	4
1.1. Lefedési feladatok megoldása lineáris progra- mozással	4
1.2. Lefedési feladatok megoldása oszlopgenerálással.	11
1.3. Lefedési feladatok megoldása 0-1 programozással.	29
1.4. Lefedési feladatok megoldása szubgradiens optimalizációval	37
2. HEURISZTIKÁRÓL ÁLTALÁBAN	46
2.1. Ládapakolási heurisztikák egydimenzióban.....	47
2.2. Heurisztikák kétdimenzióban	64
ÖSSZEFOGLALÁS	71
IRODALOMJEGYZÉK	73
FÜGGELÉK: A szabási feladat gyakorlati alkalmazása ..	78



BEVEZETÉS

Tekintsük az alábbi feladatot: Adott az $L = (a_1, a_2, \dots, a_n)$ vektor, ahol $a_i \in (0, 1]$. Helyezzük el az L elemeit minimális számú ládába úgy, hogy ne legyen olyan láda, amelyben a számok összege nagyobb 1-nél. Ez az egydimenziós láda pakolási feladat, melyet először E. ARTHURS [21]* vetett fel.

Az elnevezés is mutatja, hogy a fenti probléma nem csupán elméleti jellegű. Számos olyan ipari alkalmazás van, amely visszavezethető az előbbi feladatra. Ezek megoldására az jellemző, hogy a feladatot visszavezetik speciálisan felállított lineáris programozási /LP/ feladat megoldására. Attól függően, hogy az LP és az ezt megoldó algoritmus milyen, különböző csoportba sorolhatjuk az alkalmazásokat.

-- Több iparágban is felvetődik a selejt minimalizálásának problémája: különböző méretű félkésztermékekből vágjunk ki megrendeléseket úgy, hogy az egyes megrendelésekből a megkívánt darabszám rendelkezésre álljon, és a félkésztermékekből valamilyen értelemben a legkevesebbet használjuk fel. Az ilyen típusú feladatok megoldásával először KANTOROVICS [22] foglalkozott, kiemelve azt, hogy nagyobb méretű feladatoknál megoldása nehézségekbe ütközik. Ezt a méretekből adódó problémát küszöbölte ki GILMORE és GOMORY [9], amikor olyan megoldási módszert találtak, amelyhez nem szükséges az LP feladat összes oszlopát végigvizsgálni, hanem a bázistranszformációhoz generálják a megfelelő oszlopot. Lényegében az ő módszerüket alkalmazta TILANUS és

*

Az utalás Johnson és mtsai [21] alatti cikkében található.

GERHARDT az acéliparban [32], HAHN [15] és MADSEN [26], valamint DYSON és GREGORY [4] az üvegiparban. Papiripari alkalmazásra GILMORE és GOMORY [10] tett javaslatot.

-- Szállítással kapcsolatban merült fel az EILON és CHRISTOFIDES által megfogalmazott feladat [5]. Helyezzünk el n darab különböző méretű tárgyat dobozokban úgy, hogy az egyenként C kapacitású dobozokból a lehető legkevesebbet használjuk fel.

-- Legyen adva egy olyan szószervezésű számítógép, amelyben a szavak hossza k bit. Helyezzünk el ezekben a szavakban egy binárisan kódolt adathalmazt úgy, hogy a legkevesebb memóriát vegyük igénybe.

-- Helyezzünk el különböző méretű file-eket mágneslemezen úgy, hogy a legkevesebb területet foglaljuk el. A filekről feltecssük, hogy mindegyik rövidebb egy tracknél, és egy file nem helyezkedhet el két trackon.

-- Az órarend feladat is átfogalmazható rakodási feladattá: Adott kapacitású termekben helyezzük el a különböző hosszúságú órákat úgy, hogy a lehető legkevesebb termet használjuk fel. Az ilyen típusú feladatoknál további korlátozó feltételek is jelentkeznek.

Mivel a fenti gyakorlati problémáknak közös tulajdonságuk az, hogy egy n dimenziós halmaz elemeinek valamilyen értelemben vett optimális lefedését keressük az elemeknél kisebb - de azonos dimenzióju - elemekkel, az ilyen típusú feladatokat összefoglaló néven lefedési feladatoknak nevezzük.

Dolgozatunkban azokat az ismert módszereket tárgyaljuk, amelyek a lefedési feladatok megoldására alkalmazhatók. Ezeket az eljárásokat alapvetően két csoportra oszthatjuk:

- 1/ Egzakt megoldások, amelyek a felvetett problémák pontos - optimális - megoldását szolgáltatják esetenként bonyolult, vagy hosszan végrehajtható algoritmus segítségével.
- 2/ Heurisztikus eljárások, amelyek az esetek nagy részében csak közelítő optimumot szolgáltatnak viszonylag egyszerű algoritmus segítségével.

Ezen a helyen szeretnénk köszönetet mondani Dr.Lovász László egyetemi tanárnak és dr.Csirik János tudományos főmunkatársnak a problémák felvetéséért, a hasznos konzultációkért és azokért az észrevételekért, amelyeket a kézirat elolvasása után tettek.

Köszönettel tartozunk munkahelyi vezetőinknek a dolgozat kidolgozásához és megírásához szükséges idő biztosításáért.



1. EGZAKT MEGOLDÁSI MÓDSZEREK

Négy - elméleti alapjaiban eltérő - algoritmust ismertetünk, amelyek segítségével a lefedési feladatok megoldhatók úgy, hogy a keresett optimális megoldást megkapjuk. Az áttekinthetőbb tárgyalás kedvéért minden eljáráshoz megadunk egy alapfeladatot. Ez többnyire az a feladat lesz, amelynek kapcsán az adott megoldási módszert kidolgozták. A tárgyalásra kerülő módszerek a következők:

- Lineáris programozás /LP/
- LP oszlopgenerálással /OG/
- 0-1 programozás /O/1/
- Szubgradiens optimalizáció /SO/

1.1. Lefedési feladatok megoldása lineáris programozással

Alapfeladat: Adottak különböző hosszúságu félkésztermékek

$L_1, L_2, \dots, L_K$. Adott n különböző megrendelés $l_1, \dots, l_n$, amelyekből rendre legalább $r_1, r_2, \dots, r_n$ darabot kell készíteni a félkésztermék feldarabolásával. A feladat az, hogy a feldarabolást végezzük el úgy, hogy a lehető legkevesebb félkészterméket használjuk fel.

A fenti feladatot egydimenziós szabási feladatnak nevezzük. .

Definíció: Helyezzük el az adott L_i hosszúságú félkészterméket úgy, hogy a félkésztermék bal végpontja az origóval essen egybe. Az x pont egy vágást ad a félkészterméken, ha

$$0 \leq x \leq L_i$$

Definíció: Egy vágást valódi vágásnak nevezünk, ha

$$0 < x < L_i$$

Ez a valódi vágás a félkészterméket - vagy annak egy részét - két szakaszra bontja.

Definíció: Egy félkésztermék olyan feldarabolását, amelyben található valódi vágás, és legfeljebb egy olyan szakasza van, amelynek hossza nem egyezik meg egyetlen megrendeléssel sem, szabásnak nevezünk.

Definíció: A szabásokból alkotott halmazt szabáshalmaznak nevezünk.

Definíció: A szabáshalmaz egy részhalmazának elemeiből képzett ismétléses kombinációt szabástervnek nevezünk, ha az ebben szereplő szakaszokkal a megrendelések kielégíthetők.

Definíció: Jelöljük M_S -el az S szabástervben felhasznált félkésztermékek mennyiségét. Egy S_1 szabástervet jobb-
nak nevezünk az S_2 szabástervnél, ha

$$M_{S_1} < M_{S_2}$$



Definíció: Optimális az S_0 szabásterv akkor, ha

$$M_{S_0} \leq M_S \text{ bármely } S \text{ szabástervre.}$$

Az alapfeladatot tehát így is fogalmazhatjuk: a félkésztermékek és a megrendelések adott halmazához keressük meg az optimális szabástervet.

Definíció: Jelöljön $\underline{a}_i = (a_{1i}, \dots, a_{ni})$ egy szabást. Két szabást

- $\underline{a}_i, \underline{a}_j$ - különbözőnek nevezünk, ha van olyan k ($1 \leq k \leq n$), amelyre

$$a_{ki} \neq a_{kj}$$

Ha előállítjuk a szabáshalmazt, akkor feladatunkat a legegyszerűbben a következő LP feladat megoldására vezethetjük vissza:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1m}x_m \geq r_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2m}x_m \geq r_2$$

$\vdots$

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nm}x_m \geq r_n$$

$$x_i \geq 0 \quad i=1, \dots, m$$

$$C = c_1x_1 + c_2x_2 + \dots + c_mx_m \rightarrow \text{minimum}$$

ahol a_{ij} = az i . szabásban hányszor szerepel a j . megrendelés

c_j = a j . szabásban felhasznált félkésztermék /vagy bizonyos esetekben j . szabás/ költsége. Ez egydimenziós esetekben a legtöbbször a félkésztermék hossza, kétdimenziós esetben a területe.

m = a félkésztermékből előállítható összes különböző szabás száma /A szabáshalmaz számossága/

C = a szabásterv költsége

x_k = a k . szabás hányszor szerepel a szabástervben.

Bontsuk a költséget két részre:

- A megrendeléseket tartalmazó szakaszokra jutó költség /hasznos költség/: $C(h)$
- A félkészterméknek arra a szakaszára jutó költség, amely nem elégít ki megrendelést /selejt költség/: $C(s)$

A gyakorlati feladatoknál jelentkező költségfüggvényre többnyire érvényes a következő két feltétel:

- 1./ Additivitás $C = C(h) + C(s)$
- 2./ A hasznos költség a selejtköltség lineáris függvénye:

$$C(h) = K_1 \cdot C(s) + K_2$$

Az ilyen esetekben a fenti LP átfogalmazható úgy, hogy a feltételi rész változatlanul hagyása mellett a c_i költségek az i . szabás selejtköltségét jelentik.

Amennyiben az LP modellt kívánjuk alkalmazni, akkor a különböző korlátozó feltételeket jól kezelhetjük. Bármilyen módszerrel is oldjuk meg az LP-t, ez az előny mellett két jelentős problémával kell szembenéznünk.

a/ Az LP feladat megoldása nem biztosítja az egész értékeket.

Ezt a problémát kétféleképpen oldhatjuk meg:

-- egész értékű programozási módszert keresünk /pl. [24] ,

[30]/

-- a folytonos megoldást kerekítjük.

Ennek a megoldásnak a használhatóságára már GILMORE és GOMORY is rámutatott [9], figyelmeztetve arra, hogy csak akkor kapunk kielégítő megoldást, ha egy-egy megrendelésből a rendelt darabszám "elég magas".

b/ Miként ismert, a szimplex módszer az alapfeladathoz hasonló LP feladatokat kisegítő változók bevezetésével oldja meg. Ez azt jelenti, hogy az x_i változók mellé további új $u_1, u_2, \dots, u_n$ változókat vezet be, amelyekhez a célfüggvényben a nulla értéket rendeli. Így a módosított feladat mátrix alakja:

$$\begin{array}{l} \underline{A} \underline{X} - \underline{U} = \underline{R} \\ \underline{X}, \underline{U} \geq 0 \\ \hline \underline{C} \underline{X} + \underline{0} \rightarrow \min \end{array}$$

Definíció: Tultermelésről akkor beszélünk, ha $\exists$ olyan i ($1 \leq i \leq n$), amelyre

$$\sum_{k=1}^m a_{ik} x_k > r_i$$

Így elképzelhető, hogy minimális célfüggvényértéket akkor kapunk, ha valamely i -re tultermelés jelentkezik, hiszen az u_i segédváltozók értékei /ezek csak akkor nem nullák, ha tultermelés van/ nem befolyásolják a célfüggvény értékét. A tultermelés szabályozását az alábbi LP modell segítségével oldhatjuk meg [8]:

$$a_{11}x_1 + \dots + a_{1m}x_m - u_1 = r_1$$

$$a_{21}x_1 + \dots + a_{2m}x_m - u_2 = r_2$$

.

$$a_{n1}x_1 + \dots + a_{nm}x_m - u_n = r_n$$

$$x_i, u_i \geq 0 \quad i=1, \dots, n$$

$$c_{11}x_1 + \dots + c_{1m}x_m + c_{21}u_1 + \dots + c_{2n}u_n \rightarrow \min$$

ahol a_{ij} , x_i , r_i mint korábban

u_i = kisegítő változó

c_{1i} = az i . szabás selejtköltsége

c_{2i} = az i . szabás költsége szorozva valamilyen sullyal

A feladat fenti megfogalmazásának előnye az, hogy a segédváltozók a szimplex algoritmusban a célfüggvény értékének kiszámításakor nem zérus sullyal szerepelnek, így a súlyok megváltoztatásával a tultermelés mértéke szabályozható. Ez a szabályozási hatás jól látható az 1. ábrán, amelyet olyan megoldások alapján készítettünk, amelyeket ez utóbbi modell alkalmazásával kaptunk.

Amennyiben további korlátozó feltételeink vannak, akkor ezeket is felvehetjük az LP modell feltételeket tartalmazó részébe. Ha pl. a félkésztermékek rendelkezésre álló darabszáma van korlátozva, akkor a megoldásra alkalmas LP modell lehet a következő, amelyet [8]-ban használtunk fel:

$$\underline{\underline{A}} \underline{\underline{X}} - \underline{\underline{E}} \underline{\underline{U}} = \underline{\underline{R}}$$

$$\underline{\underline{B}} \underline{\underline{X}} \leq \underline{\underline{D}}$$

$$\underline{\underline{C}}_1 \underline{\underline{X}} + \underline{\underline{C}}_2 \underline{\underline{U}} \rightarrow \min$$

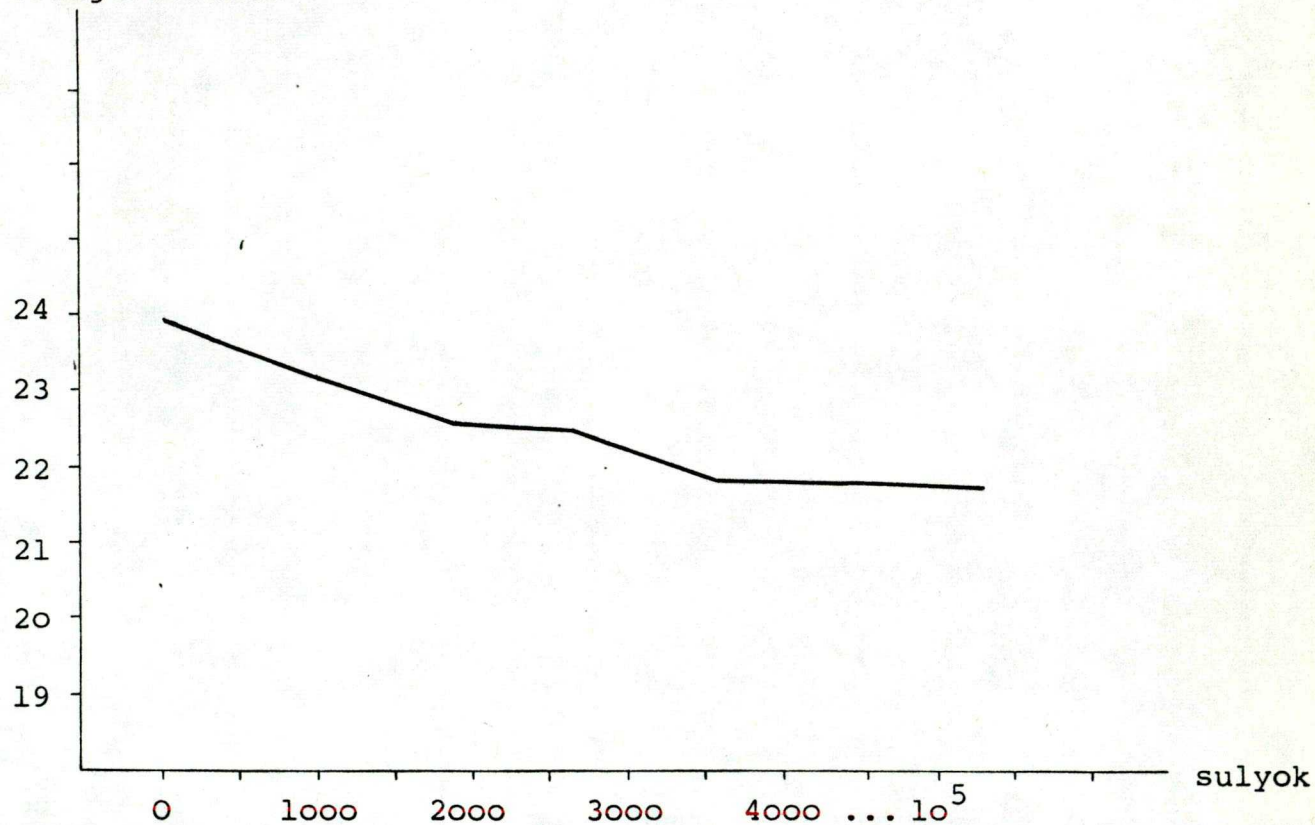
ahol $a_{ij}, x_i, u_i, r_i, c_{1i}, c_{2i}$ mint korábban

$\underline{\underline{E}}$ = egységmátrix

$b_{ij} \in \underline{\underline{B}} = \begin{cases} 1, & \text{ha az } i. \text{ szabást a } j. \text{ félkésztermékből} \\ & \text{állítottuk elő} \\ 0, & \text{különben} \end{cases}$

$d_i \in \underline{\underline{D}} = \text{az } i. \text{ félkésztermékből rendelkezésre álló mennyiség}$

selejtszázalék



1. ábra

Ennek az LP modellnek a legnagyobb hiányossága az marad, hogy nagyobb megrendelésszám esetén az összes lehetséges szabás generálása, majd a felállított LP feladat megoldása már komoly nehézségekbe ütközik. GILMORE és GOMORY [9]-ben a következő becslést adta a lehetséges szabások számára:

"200 hüvelyk hosszú félkésztermékre, 40 különböző megrendelésre, ahol a megrendelések hossza 20 és 80 hüvelyk között van, a szabások száma tullepheti a 100 milliót."

Igy az LP modellel történő megoldást akkor mondhatjuk használhatónak, ha a feladat paraméterei olyanok, hogy a lehetséges szabások száma a néhány ezret nem haladja meg. A gazdaságos alkalmazás természetesen még egyéb tényezőktől is függ /pl. futtatási gyakoriság, várható nyereség, stb./, de ezek vizsgálatára a dolgozatban nem térünk ki.

1.2. Lefedési feladatok megoldása oszlopgenerálással

Alapfeladat: Legyenek adottak L hosszúságu félkésztermékek, továbbá n különböző hosszúságu megrendelés $/l_1, l_2, l_3, \dots, l_n/$, melyekből rendre $r_1, r_2, \dots, r_n$ darabot kell készíteni. Keresük az optimális szabástervet.

Definíció: Egy $\underline{A} \underline{X} \geq \underline{R}$ LP feladatban legyen $\underline{A}$ egy $n \times m$ -es mátrix, ahol $n < m$. Legyen az $\underline{A}$ rangja n . Ekkor az $\underline{A}$ oszlopai-ból kiválasztott n darab lineárisan független vektort bázisnak nevezzük.

Felhasználva azt a tételt, hogy egy $\underline{A}$ mátrix $\underline{B}$ bázisának elemeiből lineáris kombinációval az $\underline{A}$ bármely a_k oszlopa előállítható, bevezethetjük a következő definíciót:

Definíció: Legyen

$$\underline{\underline{B}} = (\underline{a}_1, \dots, \underline{a}_n)$$

$$\underline{a}_k = x_1 \underline{a}_1 + \dots + x_n \underline{a}_n \quad k > n$$

Az előállításban szereplő x_i együtthatókat az $\underline{a}_k \in \underline{\underline{B}}$ bázisra vonatkozó koordinátáinak nevezzük.

Definíció: Jelöljük I_B -vel a $\underline{\underline{B}}$ bázishoz tartozó indexhalmazzt.

Egy $\underline{X}_0$ m dimenziós vektort az $\underline{A} \underline{X} \geq \underline{R}$ egyenlőtlenségrendszer $\underline{A}$ mátrixának $\underline{\underline{B}}$ bázisára vonatkozó megoldásának nevezzük, ha

$$1. \quad \underline{R} = x_{01} \underline{a}_1 + \dots + x_{0m} \underline{a}_m$$

$$2. \quad x_{0i} = 0, \quad \text{ha } i \notin I_B$$

Az $\underline{X}_0$ vektor azon x_{0i} komponensei, ahol $i \in I_B$, a báziskomponensek: $\underline{X}_B$.

A bázis nem egyértelműen meghatározott. Ha egyik bázisról úgy térünk át egy másikra, hogy csak egy vektorban különböznek, akkor elemi bázistranszformációról és szomszédos bázisokról beszélünk. /Elemi bázistranszformáció esetén azt mondjuk, hogy egy új elemet vontunk be a bázisba./ Bizonyítható, hogy elemi bázistranszformációkkal egy $\underline{A}$ mátrix bármely bázisa előállítható egy kiindulási bázisból.

1.2.1. TÉTEL: Legyen $\underline{\underline{B}}$ az $\underline{A} \underline{X} \geq \underline{R}$

$$\underline{X} \geq 0$$

$$\underline{C} \underline{X} \rightarrow \min$$

LP feladatban az $\underline{A}$ mátrix egy bázisa. Legyen $\underline{X}$ a $\underline{\underline{B}}$ bázishoz tartozó megoldás, továbbá $\tilde{\underline{X}}^k$ egy a bázisban nem szereplő $\underline{a}_k \in \underline{A}$ oszlop koordinátavektora.

Az $\underline{a}_k$ bevonása a bázisba akkor eredményezi a célfüggvény csökkenését, ha

$$(z_k - c_k) > 0$$

$$\text{ahol } z_k = \sum_{i \in I_B} \tilde{x}_i^k c_i$$

c_k = az $\underline{a}_k$ -hoz tartozó célfüggvényegyüttható.

Bizonyítás: Legyen $z = \sum_{i \in I_B} c_i x_i$

a $\underline{B}$ bázishoz tartozó megoldáshoz rendelt célfüggvény értéke. x_i a bázishoz tartozó megoldás egy komponense. Legyen

$$z' = \sum_{i \in I_{B'}} c_i x'_i$$

ahol $\underline{B}'$ az új bázis és x'_i a $\underline{B}'$ -höz tartozó megoldás egy komponense.

Az x'_i -k képzése a következőképpen történik /lásd [28]-t/:

$$\lambda = \min_{i \in I_B} \left(\frac{x_i}{\tilde{x}_i^k} \mid \tilde{x}_i^k > 0 \right) = \frac{x_j}{\tilde{x}_j^k}$$

$$x'_i = x_i - \lambda \tilde{x}_i^k \quad i \neq j$$

$$x'_j = \lambda \quad i = j$$

Ez azt jelenti, hogy az $\underline{a}_k$ vektort az $\underline{a}_j$ ($j \in I_B$) helyébe vonjuk be a bázisba.

Az egyszerűség kedvéért tegyük fel, hogy

$$I_B = (1, 2, \dots, n)$$

$$I_{B'} = (1, 2, \dots, j-1, j+1, \dots, n, k)$$



$$\begin{aligned}
 z' &= \sum_{i \in I_B} c_i x'_i = c_1 x'_1 + \dots + c_{j-1} x'_{j-1} + c_{j+1} x'_{j+1} + \dots + c_n x'_n + c_k x'_k = \\
 &= c_1 (x_1 - \lambda \tilde{x}_1^k) + \dots + c_{j-1} (x_{j-1} - \lambda \tilde{x}_{j-1}^k) + c_{j+1} (x_{j+1} - \lambda \tilde{x}_{j+1}^k) + \dots \\
 &+ c_n (x_n - \lambda \tilde{x}_n^k) + C_R = c_1 x_1 + \dots + c_j x_j + \dots + c_n x_n - c_1 \lambda \tilde{x}_1^k - c_2 \lambda \tilde{x}_2^k - \dots \\
 &- c_j \lambda \tilde{x}_j^k - \dots - c_n \lambda \tilde{x}_n^k + \lambda c_k = z - \lambda \sum_{i \in I_B} \tilde{x}_i^k c_i + \lambda c_k = \\
 &= z_0 - \lambda (\sum_{i \in I_B} \tilde{x}_i^k c_i - c_k) = z - \lambda (z_k - c_k)
 \end{aligned}$$

Igy a z' akkor és csak akkor lesz kisebb z_0 -nál, ha $z_k - c_k > 0$.

A bázistranszformáció ilyen végrehajtása azért hátrányos, mert ismerni kell az LP feladat minden vektorának a bázisra vonatkozó koordinátáit. Így minden új bázisra ki kell számítani ezeket a koordinátákat, és ez nagyméretű feladatoknál nagyon hosszadalmas. GILMORE és GOMORY érdeme az, hogy felismerték: nem szükséges az új bázisra vonatkozó koordinátákat az LP minden oszlopára kiszámítani, hanem a bázis ismeretében generálhatunk egy olyan oszlopot, amely a lehetséges legjobb javítást adja az LP aktuális bázisára. Elgondolásuk a következő tételen alapul:

1.2.2. TÉTEL: Legyen B az alapfeladathoz rendelt LP egy bázisa.

Jelölje $\underline{b} = b_1, \dots, b_n$ a bázishoz tartozó árnyékárakat:

$$b_k = \sum_{j \in I_B} c_j b_{jk}^{-1}$$

ahol b_{jk}^{-1} a bázis inverzének egy eleme. Legyen továbbá

$$M = \max_{a_j \in LP} \underline{b} \underline{a}_j \quad J = \max_j \underline{b} \underline{a}_j$$

Ha $\underline{a}_j$ -t bevonjuk a bázisba, akkor kapunk a kiinduló szabástervnél jobb szabástervet, ha

$$M > c_j$$

Megjegyzés: $\underline{a}_j$ egydimenziós esetben akkor szabás, ha

$$\sum_{i=1}^n l_i a_{ij} \leq L$$

Ebben az esetben a feladat megkeresni

$$M = \max_{\underline{a}_j \in LP} \left(\sum_{i=1}^n b_i a_{ij} \mid \sum_{i=1}^n l_i a_{ij} \leq L \right)$$

értékét. Ez a hátizsák feladat: adott az L befogadóképességű hátizsák és l_i méretű b_i értékű tárgyak. Töltsük fel a hátizsákot az adott tárgyakból úgy, hogy a legértékesebb legyen a tartalom.

Bizonyítás: Ha B egy bázis, akkor minden LP-beli vektor előállítható a bázis elemeinek lineáris kombinációjából:

$$\underline{a}_k = \underline{B} \underline{\tilde{x}}^k$$

ahol $\underline{\tilde{x}}^k$ az $\underline{a}_k$ vektor $\underline{B}$ bázisra vonatkozó koordinátavektora. Ekkor

$$\underline{\tilde{x}}^k = \underline{B}^{-1} \underline{a}_k \quad /1./$$

Az 1.2.1. TÉTEL-ben láttuk, hogy a javítás feltétele

$$z_k > c_k \quad \text{ahol } z_k = \sum_{i \in I_B} c_i \tilde{x}_i^k \quad /2./$$

Ezért a feltétel így is írható

$$\sum_{i \in I_B} c_i \tilde{x}_i^k > c_k \quad /3./$$

1.-ből következik, hogy

$$\tilde{x}_i^k = \sum_j b_{ij}^{-1} a_{jk}$$

Igy 3. a következő alakban írható fel:

$$\begin{aligned} \sum_{i \in I_B} c_i \tilde{x}_i^k &= \sum_{i \in I_B} c_i \sum_{j=1}^n b_{ij}^{-1} a_{jk} = \sum_{j=1}^n a_{jk} \sum_{i \in I_B} b_{ij}^{-1} c_i = \sum_{j=1}^n a_{jk} b_j = \\ &= \underline{b} \underline{a}_k > c_k \end{aligned}$$

Ez tehát azt jelenti, hogy az $\underline{a}_k$ vektort akkor vonjuk be csak a bázisba, ha

$$\underline{b} \underline{a}_k > c_k \quad /4./$$

Válasszuk tehát az összes $\underline{b} \underline{a}_k$ skalárszorzatok közül a maximálisat. Ha ennek az értéke kielégíti a 4. javítási feltételt, akkor megtaláltunk egy olyan vektort, amelyet bevonva a bázisba, megjavítjuk a célfüggvényt. Ezt az oszlopot generáló oszlopnak nevezzük.

Az alapfeladatra GILMORE és GOMORY a következő algoritmust dolgozta ki:

1. Keressünk egy kiinduló bázist.
2. Készítsük el a bázis inverzét $[\underline{B}]^{-1}$
3. Állítsuk elő az árnyékár vektort $[\underline{b}]$
4. Oldjuk meg a

$$\max_{\underline{a}_k} ([\underline{b}] \underline{a}_k \mid [\underline{1}] \underline{a}_k \leq L)$$

hátizsák feladatot.

5. Vizsgáljuk meg, hogy $M = \max_i [\underline{b}] \underline{a}_i \geq c_j$

Ha $M > c_j$, akkor az $\underline{a}_j$ vektorral hajtsunk végre egy szabályos bázistranszformációt, s ugorjunk 2.-re.

Ha $M \leq c_J$, akkor a feladatot megoldottuk, a B bázisban találhatóak azok a szabások, amelyek az optimális szabás-tervben részt vesznek.

A kérdés most már csak az maradt, hogy hogyan lehet megoldani a hátizsákfeladatot? GILMORE és GOMORY kimutatta, hogy amennyiben a következő rekurzív képlettel képezzük az

$$F_{s+1}(x) = \max_r (r b_{s+1} + F_s(x - r l_{s+1}))$$

hátizsákfüggvényeket, ahol

$$F_0(x) = 0; \quad F_s(0) = 0; \quad 0 \leq r \leq [x/l_{s+1}]; \quad x \in I; \quad a \leq L;$$

(Itt $[]$ egész részt jelöl)

Ekkor $F_n(L) = \max_i \sum b_i a_{ik}$ és a megfelelő a_{ik} -ket az adott i -hez tartozó r -ek szolgáltatják.

Láthatjuk, hogy itt a generáló oszlop kiválasztása - különösen hosszú félkésztermék esetén - bonyolult. Később a fenti szerzők [12]-ben megjavították az algoritmust. Az elmúlt évtizedben számos olyan megoldás született, amely javította az első próbálkozást. Javított algoritmust adott többek között F.A. FRIDMAN [7], INGARGIOLA [19], J.T.TENG [31], S.E.GVOZDEV [14].

Mivel a hátizsákfeladat megoldására fordított idő L méretével polinominálisan növekszik, az új módszerek keresése mellett járható út az egyszerűbb megoldásra a feladat méretének redukálása. Az ugyanis már a hátizsákfüggvények generálásából látszik, hogy nem szükséges az $F_s(x)$ értékét minden $x \leq L$ -re meghatározni, hanem elegendő azon x helyeken, ahol a félkészterméken vágási pont lehet. Így, ha valamilyen módon csökkentjük a félkésztermékből kivágható megrendelések számát, akkor csökkent-

jük a vágási pontok számát is, amivel csökken a feladat megoldására fordított idő is. Egy ilyen redukálási lehetőséget mond ki a következő tétel: [10]

1.2.3. TÉTEL: Tekintsük az

$$M = \max_{\underline{a}_k \in LP} \left(\sum_k \underline{b}_k \mid \sum_{i=1}^n l_i a_{ik} \leq L \right)$$

hátizsákfeladatot. Amennyiben van olyan $i \neq j$
- $(1 \leq i, j \leq n)$ -, amelyre

$$b_i = b_j \quad \text{és} \quad l_i > l_j$$

akkor a hátizsákfeladatnak létezik olyan megoldása,
amelyben $a_{ik} = 0$

Bizonyítás: Tegyük fel, hogy a hátizsákfeladat megoldása egy olyan $\underline{a}_J$ vektor, amelyben $a_{iJ} \neq 0$. Mivel $b_i = b_j$ és $l_i > l_j$, ezért az előző - $\underline{a}_J$ - szabáson kívül optimum lesz az az $\underline{a}'_J$ szabás is, amelyet úgy kapunk, hogy

$$a'_{1J} = a_{1J} \quad l \neq i$$

$$a'_{iJ} = 0$$

$$a'_{jJ} = a_{jJ} + a_{iJ}$$

Ez azt jelenti, hogy az i . megrendelésből keletkező vágási helyeken nem kell kiszámítani a hátizsák függvények értékét.

Az alapfeladaton túl már GILMORE és GOMORY is bevontak néhány olyan feltételt a vizsgálatba, amelyek a gyakorlati életben sűrűn fordulnak elő:

1./ Különböző hosszúságú félkésztermékek állnak rendelkezésre.

Ekkor a korábbiakban ismertetett algoritmus annyiban változik, hogy a 4. lépésben nem egy hátizsákfeladatot kell megoldani, hanem

$$\max_{a_k \in LP} (\sum b_i a_{ik} \mid \sum l_i a_{ik} \leq L_j) \quad j=(1, \dots, K)$$

alaku feladatot oldunk meg mindaddig, amíg az 5. lépésben $< c_j$ -t nem találunk. Ebben az esetben felvetődik a kérdés, miként a szimplex módszernél is, hogy mi a gazdaságosabb: az először megtalált javítóoszlopot beépíteni a bázisba, vagy keressük meg azokat az a_j -ket, amelyek javítást adnak, és ezek közül válasszuk ki azt, amelyik a célfüggvény legjobb javítását adja? GILMORE és GOMORY kísérleti eredményei [10] azt bizonyítják, hogy az utóbbi módszer gyorsabb, tehát érdekesebb

$$\max_J (\max_{a_k \in LP} \sum b_i a_{ik} - c_j)$$

értékét kiszámítani és erre a félkésztermékre vonatkozó megoldást bevonni a bázisba.

2./ Vágófejek számának korlátozása.

Gyakorlati feladatoknál - az automatizálás következtében - előfordulhat, hogy a félkészterméken nem végezhetünk akárhány vágást, hanem a vágások száma korlátozva van. Tegyük fel, hogy R darab vágófej használható. Ez azt jelenti, hogy a hátizsákfeladat megoldásakor a "hosszfeltételen" túl figyelembe kell venni egy újabb feltételt is. Így a megoldandó hátizsákfeladat:

$$\max_{a_k \in LP} (\sum_{i=1}^n b_i a_{ik} \mid \sum_{i=1}^n l_i a_{ik} \leq L, \sum_{i=1}^n a_{ik} \leq R)$$

3./ Gép súlyozási probléma



Tekintsünk egy olyan papíripari feladatot, amelyben a gépek különböző megrendelésekre alkalmazhatók csak. Az összes megrendelés nem vágható le egy gépen akkor sem, ha az adja a legkisebb veszteséget, mivel a felvágható mennyiséget az idő korlátozza. Ezért szükség van arra, hogy a szabásokat úgy végezzük el, hogy az egyik gép ne álljon feleslegesen akkor, amikor a másikat túlterhelnénk.

Legyen P darab gépünk úgy, hogy az r . gép Q_r darab L_r hosszúságú tekercset tud felválni adott időszak alatt. Ekkor az alábbi LP feladatra vezethetjük vissza a megoldást:

$$\begin{aligned} \sum_j \sum_r a_{ijr} x_{jr} &\geq n_i & i = 1, \dots, n \\ \sum_j x_{jr} &\leq Q_r & r = 1, \dots, P \end{aligned}$$

$$\sum_j \sum_r c_r x_{jr} \rightarrow \min$$

ahol a_{ijr} = az r . gépre vonatkozó szabásban hányszor fordul elő az l_i hosszúságú megrendelés.

n_i = hány darabot kell leszabni az i . megrendelésből.

x_{jr} = az r . gépen a j . szabást hányszor alkalmazzuk az optimális szabástervben.

c_r = az r . gépen felvágható félkésztermék költsége.

Ezek után keressük azt az oszlopot, amely javítást ad a célfüggvényben egy kiindulási bázishoz viszonyítva. A választásnál vegyük figyelembe, hogy $\underline{a}_k$ generáló oszlop most két részből áll:

$$\underline{a}_k = (\underline{a}_k^1, \underline{a}_k^2)$$

ahol $\underline{a}_{ik}^1$ jelentése megegyezik a korábbi a_{ik} -k jelentésével, míg

$$a_{ik}^2 = \begin{cases} 1, & \text{ha } i = r \text{ az } r. \text{ gépre készült szabásnál.} \\ 0, & \text{különben} \end{cases}$$

Ekkor a hátizsák feladat a következő lesz:

$$\max_{a_k \in LP_r} (c_r - \sum_{i=1} b_i a_{ik}^i + b_{n+r} \mid \sum_{i=1} l_i a_{ik}^i \leq L_r)$$

Most P db különböző hátizsák feladatot kell megoldanunk és ezek maximumát tekintjük generáló oszlopnak.

Az algoritmus ezek után megegyezik a korábban leirtakkal. Az olyan gépsúlyozási feladat, amelyben alsó és felső korlátok vegyesen fordulnak elő, hasonlóan oldható meg. Megjegyezzük, hogy az olyan selejtminimalizálási feladat, amelyben a félkésztermékek rendelkezésre álló darabszáma korlátozott, hasonló LP feladathoz vezetnek.

4./ Felhasználói tűrések megengedése.

Itt a megrendelések olyanok, hogy nem pontosan kell őket kielégíteni: amennyiben a megrendelő n_i darabot rendel az l_i hosszúságu anyagból, akkor ez az ilyen jellegű feladatoknál azt jelenti, hogy " n_i körüli" darabszámra van szüksége. A megengedett tűrés általában $\pm 5\%$.

Azt az esetet fogjuk vizsgálni, amikor csak egyfajta félkésztermék van $/L/$. A feladat megoldásában az első nehézséget az okozza, hogy nem elegendő a teljes veszteséget minimalizálni, mert az általában növekedni fog, ha kihasználjuk az egyenlőtlenséggel adott intervallumot:

$$n_i' \leq n_i \leq n_i''$$

Mivel a hatékonyság mértékét legjobban a százalékos veszteség fejezi ki, ezért célszerű új célfüggvényt bevezetni. Rendeljük a j . vágási képhez a következő veszteséget:

$$w_j = L - \sum_{i=1}^n l_i a_{ij}$$

Legyen az új célfüggvényünk:

$$\zeta = \frac{z_1}{z_2} = \frac{\sum_{j=1}^m w_j x_j}{\sum_{j=1}^m x_j} \rightarrow \min$$

Az LP feladat feltételrendszere pedig a következő alakban írható fel:

$$\sum_{j=1}^n a_{ji} x_j - s_j = n'_i$$

$$0 \leq s_j \leq n'_i{}' - n'_i$$

Látható, hogy a feltételrendszerben nincs különösebb változás. A probléma a célfüggvényben van. Ez ugyanis nem lineáris. MARTOS B. [27] mutatta ki, hogy az ilyen típusu - hiperbolikus - feladat visszavezethető lineáris célfüggvényre, ha az LP feladatot további két változóval és két új egyenlőtlenséggel bővítjük.

$$z_1 = \sum_{j=1}^m w_j x_j \geq 0$$

$$z_2 = \sum_{j=1}^m x_j \geq 0$$

Igy az LP feladat egy oszlopát a következőképpen jelölhetjük:

$$\underline{a}_j = (-w_j, -1, a_{1j}, \dots, a_{nj})$$

/Itt azt feltételezzük, hogy a két új feltétel az LP feladat első két sorában helyezkedik el./

GILMORE és GOMORY bebizonyította, hogy ilyen esetben is visszavezethető az LP feladat megoldása hátizsák függvények megoldására. Bizonyításuk a következő:

Tegyük fel, hogy számításaink során eljutottunk egy olyan bázis inverzhez, amelynek első két sora:

$$\underline{B}_1 = (1, 0, \underline{b}^1)$$

$$\underline{B}_2 = (0, 1, \underline{b}^2)$$

ahol $\underline{b}^1$ és $\underline{b}^2$ n dimenziós vektorok. Jelöljük ennek a két vektornak egy tetszőleges oszloppal $\underline{a}_j$ való szorzatát:

$$-\frac{dz_1}{dx_j} = \underline{a}_j \underline{B}_1$$

$$-\frac{dz_2}{dx_j} = \underline{a}_j \underline{B}_2$$

Mivel

$$\zeta = \frac{z_1}{z_2}$$

ezért

$$\frac{d\zeta}{dx_j} = \left(\frac{dz_1}{dx_j} z_2 - \frac{dz_2}{dx_j} z_1 \right) / z_2^2 =$$

$$= [z_1 (0, 1, \underline{b}^2) \underline{a}_j - z_2 (1, 0, \underline{b}^1) \underline{a}_j] / z_2^2 =$$

$$= [z_1 \left(\sum_{i=1}^n b_i^2 a_{ij} - 1 \right) - z_2 \left(\sum_{i=1}^n b_i^1 a_{ij} - w_j \right)] / z_2^2 =$$

$$= [z_2 w_j - z_1 + \sum_{i=1}^n a_{ij} (b_i^2 z_1 - b_i^1 z_2)] / z_2^2 =$$

$$= [z_2^L - z_1 + \sum_{i=1}^n a_{ij} (b_i^2 z_1 - b_i^1 z_2 - z_2^1 l_i)] / z_2^2 =$$

Legyen $k = z_2^L - z_1$

$$\bar{b}_i = b_i^2 z_1 - b_i^1 z_2 - z_2^1 l_i$$

Igy

$$\frac{d\zeta}{dx_j} = (k - \sum_{i=1}^n a_{ij} \bar{b}_i) / z_2^2$$

Ez a kifejezés akkor negatív, ha x_j növekedése csökkenteni fogja a célfüggvény értékét. Oszlopot tehát úgy kell választanunk, hogy a legnegatívabb $d\zeta/dx_j$ értéket keressük. Ez nem jelent mást, mint minimalizálni a

$$k - \sum_{i=1}^n a_{ij} \bar{b}_i$$

kifejezést. Ezt pedig a következő hátizsák feladat segítségével kaphatjuk meg:

$$\max_{a_j \in LP} \left(\sum_{i=1}^n \bar{b}_i a_{ij} \mid \sum_{i=1}^n l_i a_{ij} \leq L \right)$$

Ennek megoldása után áttérhetünk az eredeti algoritmusra.

Selejtminimalizálási feladat két dimenzióban

Alapfeladat: adottak (L, W) - bonyolultabb esetben (L_i, W_i)

$(i=1, \dots, K)$ - méretű téglalap alaku félkésztermékek, amelyekből meghatározott számú (l_i, w_i) méretű téglalap alaku megrendeléseket kell kiszabni úgy, hogy a felhasznált félkésztermékek összterülete minimális legyen.

Helyezzük el a téglalap alakú, (L, W) méretű félkészterméket egy derékszögű koordináta-rendszerben úgy, hogy a sarokpontok koordinátái rendre $(0, 0), (0, L), (L, W), (L, 0)$ legyenek.

Definíció: Az (x_k, y_k) és (x_v, y_v) pontokat összekötő szakaszt az (L, W) méretű félkészterméken egy vágásnak nevezzük, ha

$$0 \leq x_k \leq x_v \leq L$$

$$0 \leq y_k \leq y_v \leq W$$

Definíció: Egy vágást horizontálisnak nevezzük, ha

$$x_k \leq x_v \text{ és } y_k = y_v$$

Definíció: Egy vágást vertikálisnak nevezzük, ha

$$x_k \leq y_v \text{ és } x_k = x_v$$

Definíció: Guillotine vágásról csak horizontális vagy vertikális vágások esetén beszélünk. Ha az (X_0, Y_0) méretű félkészterméket - vagy annak guillotine típusú vágásokkal keletkezett darabját - úgy vágjuk, hogy

- horizontális vágásnál $x_k = 0$ és $x_v = X_0$
- vertikális vágásnál $y_k = 0$ és $y_v = Y_0$

Definíció: Egy félkésztermék feldarabolását szabásnak nevezzük, ha a benne szereplő vágások legalább egy megrendelést meghatároznak.

Definíció: Jelöljük S_i -vel az i . szabásban lévő olyan területek összegét, amelyek nem határoznak meg megrendelést. Legyen ez az i . szabás selejtje.

Definíció: Az i . szabást jobbnak nevezzük a j . szabásnál, ha

$$S_i < S_j$$

Definíció: Optimális az az i_0 szabás, amelyhez nem adható meg olyan j . szabás, amelyhez tartozó

$$S_j < S_{i_0}$$

Mivel az általános eset tárgyalása rendkívül bonyolult, ezért csak olyan esetekkel foglalkozunk, amikor a szabásban csak guillotine típusu, horizontális és vertikális vágások fordulnak elő.

Kézenfekvőnek látszik, hogy az egydimenziós esethez hasonlóan járjunk el. Ha a feladat paraméterei megengedik, akkor készítsük el az összes szabást. Ebben az esetben a probléma az, hogy nagyon nehéz felső becslést adni az összes lehetséges szabások számára, és így nehéz megbecsülni az LP feladat méreteit. Ha figyelembe vesszük a következő könnyen belátható tételt, akkor már könnyebb becslést adni:

1.2.4. TÉTEL: Az optimális szabások halmazából készíthető optimális szabásterv.

Bizonyítás: Legyen S_0 egy olyan szabásterv, amely optimális, és tartalmaz nem optimális szabást. Ez a szabás kicserélhető egy olyan optimális szabással, amelyből ugyanannyi, vagy nála kevesebb szükséges az optimális szabástervhez. Így a célfüggvény értéke nem növekszik a helyettesítéssel.

N. CHRISTOFIDES és C. WHITLOCK [3] iteratív, J.C.HERZ [18] rekurzív algoritmust készített, amely adott megrendelésállomány figyelembe vételével elkészíti egy félkésztermék optimális sza-

bását. Ennek figyelembe vételével készíthetünk olyan algoritmust, amelyben a szabások száma meghatározható. Legyen n különböző megrendelésünk. U_i tartalmazza azokat az optimális lefedéseket, amelyeket i . darab megrendelés segítségével készítettünk.

1. $k = 0$
2. Állítsuk elő a Chrostofides-Whitlock vagy a Herz féle algoritmus segítségével az U_{n-k} szabáshalmaz optimális szabásait.
3. $k = k+1$. Ha $k \leq n-1$, akkor ugorjunk 2-re.
Ha $k = n$, akkor ugorjunk 4-re.
4. Előállítottuk az összes optimális szabást. Készítsük el a feladathoz az LP-t és oldjuk meg.

A feladatban szereplő szabások száma:

$$m = \sum_{i=1}^n n^i = \frac{n^{n+1} - 1}{n-1}$$

Mint látható, az LP-ben található szabások száma a megrendelések számával exponenciálisan növekszik. Ezért egy másik lehetséges megoldás, hogy az optimális szabástervet oszlopgenerálás segítségével keressük meg. Belátható, hogy az alap-algoritmus nem változik, csak a hátizsákfüggvény értékeinek a kiszámítása változik.

Guillotine vágások esetére GILMORE és GOMORY a következő rekurziót javasolta a hátizsák függvények kiszámítására [10]:

$$G(x, y) = \max_{\substack{0 \leq x_0 \leq \lfloor x/2 \rfloor \\ 0 \leq y_0 \leq \lfloor y/2 \rfloor}} (g(x, y), G(x_0, y) + G(x-x_0, y), G(x, y_0) + G(x, y-y_0))$$

$$\text{ahol } g(x,y) = \max_{\substack{w_i \leq x \\ l_i \leq y}} b_i$$

Egy másik, jól kezelhető csoportját képezik a kétdimenziós lefedési feladatoknak azok a problémák, amelyeknél a szabásban csak két ütemű vágásokat engedünk meg. Ilyenkor a szabás kialakítása két lépésben történik:

- először guillotine vágásokkal csikokra vágjuk a félkészterméket, majd
- a csikokat guillotine vágásokkal a kívánt méretre vágjuk az előző vágási irányra merőlegesen.

Ha a felvágott csikon belül megengedünk még egy harmadik - az első vágással párhuzamos - vágást, akkor nem egzakt, ellenkező esetben egzakt esetről beszélünk. Nem egzakt esetre a hátizsák feladat megoldása a következő:

- 1./ Keressük meg a (w_i, L) méretű félkésztermék optimális szabását, azaz, oldjuk meg a

$$b'_i = \max_k \left(\sum_{j=1} b_j a_{jk} \mid \sum_{j=1} l_j a_{jk} \leq L, w_j \leq w_i \right)$$

hátizsák feladatot minden w_i szélességre. A megoldásként kapott $\underline{a}_k$ -k fogják szolgáltatni a generáló csikszabásokat. Jelöljük $\underline{a}_i$ -vel a w_i csikszélességhez tartozó generáló oszlopot.

- 2./ Az optimális csikszabásokból állítsunk elő generáló szabást. A megoldást a következő hátizsákfüggvény segítségével kapjuk meg:

$$\max_j \left(\sum_{i=1} b'_i d_{ij} \mid \sum_{i=1} w_i d_{ij} \leq W \right)$$

Legyen a megoldás $\underline{d}_J$. Ez azt jelenti, hogy az $\underline{a}_i$ csikvektor a generáló szabásban $\underline{d}_{iJ}$ -szer fog szerepelni.

Ugy tűnik, hogy a feladat megoldása tulságosan bonyolult. Mégis kimutatható, hogy az algoritmusban ciklusonként mindössze két hátizsákfeladatot kell megoldani. Tegyük fel ugyanis, hogy $w_i \leq w_j$ ha $i < j$. Ekkor

$$F_{w_i}(L) = b'_i \quad /1./$$

Ez azt jelenti, hogy az első lépésben - akárhány lehetséges csikszélesség is van - elegendő egyetlen hátizsákfeladatot megoldani, hiszen $/1/$ értékeket $F_n(L)$ meghatározásához a rekurzív formula miatt mindenképpen ki kell számítani.

1.3. Lefedési feladatok megoldása 0-1 programozással

Alapfeladat: Legyen adva n darab egydimenziós tárgy, amelyek Q_i -vel mérhetők ($i=1, \dots, n$). Helyezzük el a tárgyakat C kapacitású dobozokban úgy, hogy a kapacitáskorlátokat ne lépjük túl és a lehető legkevesebb dobozt használjuk fel.

A feladat megoldását egy olyan LP feladat segítségével fogjuk megoldani, amelyben a változók csak 0 és 1 értékeket vehetnek fel. Jelölje $X = (x_{ij})$ azt az allokációs mátrixot, amely megmutatja, hogy melyik tárgyat melyik dobozba helyezzük:

$$x_{ij} = \begin{cases} = 1, & \text{ha az } i. \text{ tárgy a } j. \text{ dobozban van} \\ = 0 & \text{különben} \end{cases}$$

A kapacitáskorlátokat a következő egyenlőtlenségrendszerrel vehetjük figyelembe:

$$\sum_{i=1}^n Q_i x_{ij} \leq C \quad j=1, \dots, N$$

ahol $0 \leq Q_i \leq C$

Mivel minden tárgyat egy és csak egy dobozba allokálhatunk:

$$\sum_{j=1}^N x_{ij} = 1$$

Rendeljünk egy V_j büntetést minden olyan j . dobozhoz, amelybe már allokáltunk tárgyat. V_j -k legyenek olyanok, hogy

$$V_{j+1} \gg V_j > 0$$

Ekkor egy büntetésfüggvényt definiálhatunk az allokáláshoz:

$$z = \sum_{j=1}^N (V_j \sum_{i=1}^n x_{ij})$$

A V_j -k megfelelő választása mellett az allokálás olyan lesz, hogy a magasabb indexű dobozba csak akkor helyezünk el tárgyat, ha az alacsonyabb indexűekbe már ne lehet.

A kérdés az, hogy hogyan definiáljuk a V_j -ket? Egy lehetséges generálást adott EILON és CHRISTOFIDES [5]-ben: tegyük fel, hogy maximum p darab tárgy helyezhető el egy dobozban:

$$p = \left\lfloor \frac{C}{Q_0} \right\rfloor \quad \text{ahol } Q_0 = \min_i Q_i \text{ és } \lfloor \cdot \rfloor \text{ egész részt jelöl.}$$

Legyen $V_1 = 1$

és $V_{j+1} > V_j p \quad j \geq 2$

Feladatunkat tehát a következőképpen foglalhatjuk össze:

$$\sum_{i=1}^n Q_i x_{ij} \leq C$$

$$\sum_{j=1}^N x_{ij} \leq 1$$

$$z = \sum_{j=1}^N (V_j \sum_{i=1}^n x_{ij}) \rightarrow \min$$

Gyakorlati alkalmazásokban további korlátozó feltételek léphetnek fel:

- az i . tárgyat nem tehetjük a j . dobozba:

$$x_{ij} = 0$$

- két vagy több tárgy nem allokálható ugyanabba a dobozba.

Legyen a két tárgy i_1 , és az i_2 .

$$x_{i_1 j} + x_{i_2 j} = 1$$

A feladatot megoldhatjuk akár szimplex módszer, akár Gilmore-Gomory módszer segítségével, azonban a speciális változók miatt egyszerűbb módszer is célhoz vezet.

Két algoritmust ismertetünk a megoldásra:

- a/ leszámlálási algoritmus
- b/ Branch and bound algoritmus

a/ Leszámlálási algoritmus

Az eljárás lényege, hogy valamilyen módszer szerint bejárjuk a döntési fa minden ágát, és a legjobb megoldást őrizzük meg. Itt egy rekurzív generáláson alapuló módszerrel ismertetünk:



1./ Valamilyen heurisztikus algoritmus /pl.[5]/ segítségével keressünk egy közelítő megoldást. Legyen ez $\bar{z}$.

2./ Legyen $L = (1, 1, \dots, 1)$ egy olyan n elemű vektor, amelynek k . eleme azt jelenti, hogy a k . tárgyat hányadik dobozba helyeztük. Így az L segítségével definiált megoldásra teljesül, hogy

$$\sum_{j=1}^N x_{ij} \leq 1$$

3./ Vizsgáljuk meg, hogy az L -hez rendelt $\underline{X} = (x_{ij})$ megoldásmátrix lehetséges megoldás-e, azaz teljesül-e, hogy

$$\sum_{i=1}^n x_{ij} Q_i \leq C \quad 1 \leq j \leq N$$

4./ Ha $\underline{X}$ nem lehetséges megoldás, akkor ugorjunk 8-ra.

5./ Ha $\underline{X}$ lehetséges megoldás, akkor vizsgáljuk meg, hogy a hozzá tartozó

$$z = \sum_{j=1}^N (V_j \sum_{i=1}^n x_{ij})$$

célfüggvényértékre

$$z < \bar{z}$$

teljesül-e.

6./ Ha $z \geq \bar{z}$, akkor ugorjunk 8-ra.

7./ Ha $z < \bar{z}$, akkor helyettesítsük $\bar{z}$ -t z -vel és legyen az új, aktuális megoldás X .

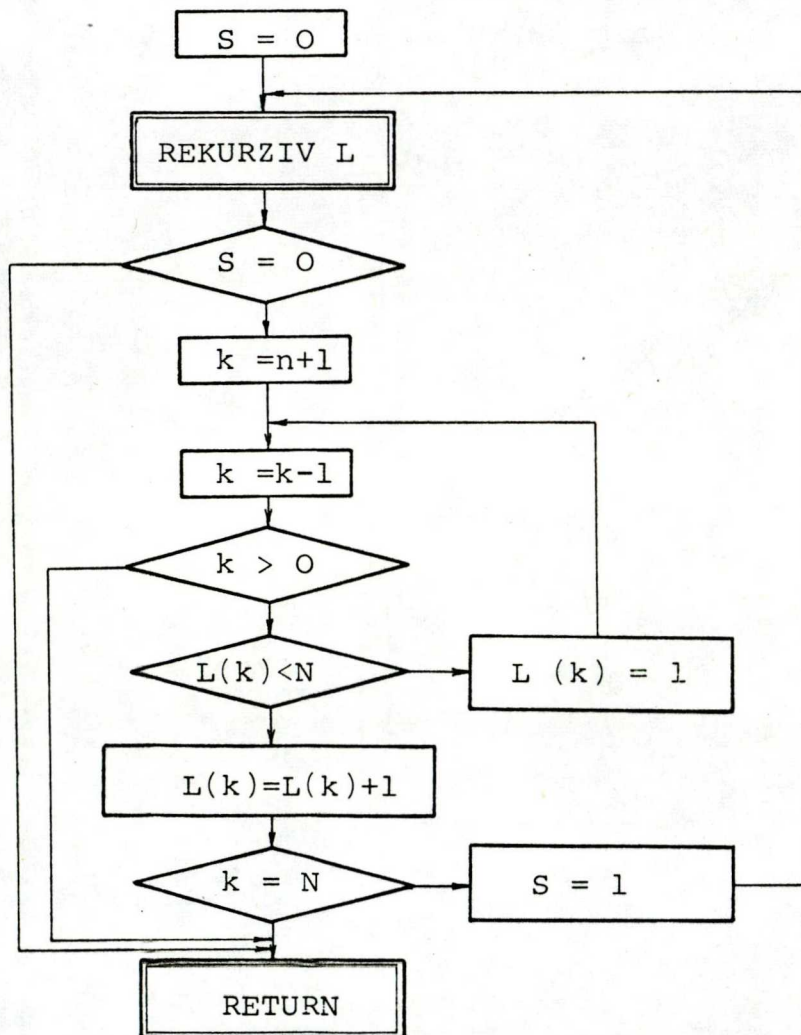
8./ A rekurzív algoritmus /lásd 2. ábra/ segítségével határozzuk meg az új L -t

9./ Ha $L \neq (1, 1, \dots, 1)$, akkor ugorjunk 3.-ra, különben bejártuk a fa minden ágát, $\bar{z}$ az optimális célfüggvényérték, a hozzá tartozó X az optimális megoldás.

n = a tárgyak száma

N = a dobozok száma

L = n elemű vektor, amelynek k . eleme azt jelenti, hogy a k . elemet hányadik dobozba tettük.



2. ábra

A rekurzív szubrutin folyamatábrája

b/ Branch and bound algoritmus

A módszer azon alapul, hogy bizonyos ágak vizsgálatát egy adott ponttól ki lehet zárni egy előzetes ellenőrzés után. Egy ilyen algoritmust közölt EILON és CHRISTOFIDES [5].

A módszer alapja, hogy az nN darab x_{ij} változót egy S halmazba helyezzük el. Ezek közül k darabot rögzítünk, a megmaradókat $- nN - k$ darab - szabad változóknak nevezzük.

E szabad változók egy alproblémát alkotnak, amelyet

- vagy megoldunk úgy, hogy meghatározzuk a szabad változókra a legjobb célfüggvény értékét. Ezeket egyesítve az S halmaz elemeivel megvizsgáljuk, hogy lehetséges megoldást szolgáltatnak-e, és az egyesített halmazhoz tartozó célfüggvényérték kisebb-e mint az eddigi minimum;
- vagy nem találunk lehetséges megoldást.

Ezek után k -t csökkentve megismételjük az eljárást. Ha a kapott megoldás nem lehetséges, akkor egy T halmazt definiálunk. Ez a halmaz tartalmazni fogja azokat a csomópontokat, amelyek a szabad változókra előfordulhatnak. T -nek egy elemét tegyük át S -be. Ezzel egy új alproblémát definiálunk. T -nek ezt az elemét válasszuk úgy ki, hogy a legnagyobb Q_i kiterjedésű tárgyatallokáljuk abba a dobozba, amelynek a legkisebb a ki nem használt kapacitása.

Ahhoz, hogy kizárjuk a fának ezt a részét eddig a pontig, oldjunk meg egy LP feladatot oly módon, hogy a változókat folytonosnak tekintjük a $[0,1]$ intervallumon. Amennyiben az LP megoldása $\bar{z}$ -nél nagyobb, akkor az egész ágot kizárhatjuk. Ha nem, akkor tovább kell vizsgálni az ágot. Az algo-

ritmus a következő:

1. Kezdeti érték meghatározása $\bar{z}$ -ra /pl. heurisztika/
2. Legyen $S = \{0\}$, $x_{ij}=0$.
3. Keressük meg a lehetséges csomópontok T halmazát.
4. Ha T üres, akkor ugorjunk 9-re, különben 5-re.
5. Tegyük át egy elemet T -ből S -be.
6. Keressük meg az $\underline{x}$ megoldást és a hozzá tartozó célfüggvény értéket.
7. Ha $\underline{x}$ lehetséges megoldás, akkor ugorjunk 8-ra, különben 3-ra.
8. Ha $z < \bar{z}$, akkor legyen $\bar{z} = z$ és $\underline{x}$ az új, aktuális megoldás.
9. Ellenőrizzük S -t visszafele. Vizsgáljuk meg, hogy van-e olyan változó, amelyet a komplementisével helyettesítve a keletkező probléma olyan, hogy azt még nem oldottuk meg.
10. Ha találtunk ilyen változót, akkor helyettesítsük a komplementisével, és ugorjunk 6-ra. Ellenkező esetben megtaláltuk azt a $\bar{z}$ értéket, amelyhez rendelt megoldás a legjobb pakolást biztosítja.

Az alapfeladatban felírt problémát két irányban lehet kiterjeszteni:

- többdinemziós korlátok
- változó kapacitású dobozok

Rakodási feladat többdimenziós korlátokkal

A gyakorlati életben többnyire olyan problémák jelentkeznek, ahol olyan feladatokat kell megoldani, amelyekben nem egyfajta korlát van. Ugyanabban a feladatban korlát lehet pl. a súly, az érték, a térfogat, stb.

Általában egy tárgyat r értékkel specifikálhatunk. Hasonlóan minden doboznak r kapacitáskorlátja van. A megoldandó feladat így annyiban módosul, hogy a kapacitáskorlátokat r -szer kell megismételni:

$$\sum_{i=1}^n Q_{ik} x_{ij} \leq C_k \quad 1 \leq k \leq r$$

A 0-1 programozási feladat megoldása nem kíván semmilyen módosítást.

Rakodási feladat különböző kapacitású dobozokkal

Az ilyen típusu feladat modelljében két jelentős változás lesz az alapfeladathoz viszonyítva:

- Mivel a különböző kapacitásokat egy vektorban kell megadni, ezért meg kell adni a dobozok számát is;
- Nem használhatjuk a célfüggvény értékeinek a kiszámítására a büntetés súlyokat, mert a súlyok nem határozhatók meg egyértelműen. Legyen $X = x_{ij}$ az allokációs mátrix, továbbá

$$y_i = \begin{cases} 1, & \text{ha az } i. \text{ dobozba már pakoltunk} \\ 0, & \text{ha az } i. \text{ doboz teljesen üres} \end{cases}$$

L jelöljön egy n -nél nagyobb tetszőleges valós számot.

N jelölje a C_j $j = (1, \dots, N)$ kapacitású dobozok számát.

Ekkor a modellünk:

$$\sum_{j=1}^N x_{ij} = 1$$

$$\sum_{i=1}^n Q_i x_{ij} \leq C_j \quad j = 1, \dots, N$$

$$\sum_{i=1}^n x_{ij} \leq Ly_j \quad \times$$

$$z = \sum_{j=1}^N y_j \rightarrow \min$$

A $\times$ feltétel azt biztosítja, hogy olyan dobozba, amelyre $y_j = 0$, ne legyen tárgy.

Természetesen lehetséges az egyenlő kapacitású dobozok problémájára az utóbbi modellt felállítani, azonban ekkor a bevezetett y_j változók miatt a feladat mérete - és ezzel a megoldási ideje is - növekedne.

1.4. Lefedési feladat megoldása szubgradiens optimalizációval

A szubgradiens optimalizációt akkor próbálták meg először lefedési feladatok megoldására is felhasználni, amikor HELD és KARP [16] sikeresen alkalmazta nagyméretű utazó ügynök probléma megoldására.

Tekintsük a következő általános matematikai programozási feladatot /ÁLP/: Legyenek adottak $\underline{z}_k = (z_{k1}, \dots, z_{kn})$ n dimenziós vektorok $/k = 1, \dots, n/$. Tartozzon minden $\underline{z}_k$ -hoz egy valószínűségi skalár. Keressük azt az $\underline{y} = (y_1, y_2, \dots, y_n)$ vektort, amelyre a $w = w(\underline{y})$ függvény maximális, ahol

$$w(\underline{y}) = \min_k (c_k + \underline{y}_k \underline{z}_k) \quad \underline{y} \in Q$$

$Q \subseteq R^n$ konvex $/\underline{y}$ -ra nem kötjük ki feltétlen a korlátosságot./



1.4.1. TÉTEL: Minden olyan feladat, amely a Dantzig-Wolfe algoritmussal megoldható, helyettesíthető egy olyan LP feladattal, amelynek alakja megegyezik az ÁLP alakjával. A két LP ugyanazt a megoldást szolgáltatja.

Bizonyítás: Jelöljük a Dantzig-Wolfe algoritmussal megoldható feladatot

$$\begin{aligned}\underline{A}\underline{X} &\leq \underline{B} \\ \underline{A}'\underline{X} &= \underline{B}' \\ \underline{X} &\geq 0\end{aligned}$$

$$\underline{C}\underline{X} \rightarrow \min$$

ahol feltételezzük, hogy az $\{\underline{X} \mid \underline{A}'\underline{X} = \underline{B}', \underline{X} \geq 0\}$ halmaz az $\underline{X}^k$ extrémális pontokkal van korlátozva. Mivel minden olyan $\underline{X}$ -re, amely az előbbi halmazban benne van igaz, hogy

$$\underline{X} = \sum_{k=1}^n \lambda_k \underline{X}^k, \text{ ahol } \sum_{k=1}^n \lambda_k = 1$$

ezért az LP feladat felírható

$$\begin{aligned}\sum_{k=1}^n \underline{A}\underline{X}^k \lambda_k &= \underline{B} \\ \sum_{k=1}^n \lambda_k &= 1 \\ \lambda_k &\geq 0\end{aligned}$$

$$\sum_{k=1}^n \underline{C}\underline{X}^k \lambda_k \rightarrow \min$$

alakban. Irjuk fel a fenti primál LP feladathoz tartozó duálist:

$$\underline{Y} (\underline{AX}^k) + \underline{R} \leq \underline{CX}^k$$

$$\underline{YB} + \underline{R} \rightarrow \max$$

Mivel $\underline{R} \leq \underline{CX}^k - \underline{Y}(\underline{AX}^k)$, ezért a duális feladat célfüggvénye felírható

$$\underline{YB} + \underline{R} \leq \underline{CX}^k - \underline{Y} (\underline{AX}^k - \underline{B})$$

alakban. Jelöljük

$$w(Y) = \min_k (\underline{CX}^k - \underline{Y} (\underline{AX}^k - \underline{B})) \quad *$$

Ekkor a duális feladat megfogalmazható a következőképpen:
keressük meg a $w(\underline{Y})$ függvény maximumát, ha

$$\max_{w, Y} \{w \mid w \leq \underline{CX}^k - \underline{Y} (\underline{AX}^k - \underline{B})\}$$

és $w(Y)$ -ra teljesül a $*$ feltétel. Ez viszont éppen az ÁLP feladat.

Ha m viszonylag kicsi és Q egy konvex poliéder, akkor az ilyen formájú feladatot meg lehet oldani a simplex módszer segítségével. Egy másik lehetséges megoldás a szubgradiens optimalizáció.

Legyen $Q = E^n$. Ekkor a kiindulási ÁLP feladat:

$$\max_{Y \in Q} \{w(\underline{Y}) \mid w(\underline{Y}) = \min_k (c_k + \underline{Y} \underline{z}_k)\}$$

Tegyük fel, hogy $w(Y)$ korlátos felülről. Mivel w szakaszonként lineáris, tehát folytonos, ezért van olyan Y' hely, ahol a függvény felveszi maximumát:

$$w(\underline{Y}') = w' = \max_Y w(\underline{Y})$$

Legyen $Y \in Q$. Ekkor létezik legalább egy olyan k , amelyre $w(Y)$ felveszi a minimumát.

Legyen

$$\underline{V}(\underline{Y}) = \{\underline{z}_k \mid c_k + \underline{Y} \underline{z}_k = w(Y), \quad k=1, \dots, m\}$$

$\underline{V}(\underline{Y})$ általában egy elemet tartalmaz. Kimutatható, hogy ekkor a $w(\underline{Y})$ függvény differenciálható az $\underline{Y}$ pontban. és a $\underline{z}_k \in \underline{V}(\underline{Y})$ éppen a $w(\underline{Y})$ függvény gradiensé: $\nabla w(\underline{Y})$. Ha $\underline{V}(\underline{Y})$ nem egy elemű, akkor w nem differenciálható az $\underline{Y}$ -ban.

Definíció: Az $\underline{u} = (u_1, \dots, u_n)$ vektort akkor nevezzük az $f(\underline{X})$ függvény $\underline{X}_0$ pontbeli szubgradiensének, ha

$$f(\underline{X}) - f(\underline{X}_0) \leq \underline{u} (\underline{X} - \underline{X}_0) \quad \forall (\underline{X}) \neq \underline{X}_0; \quad \underline{X} \in Q.$$

Megjegyzés: ha $f(\underline{X})$ differenciálható, akkor

$$\underline{u} = \nabla f(\underline{X})$$

1.4.2. TÉTEL: Jelöljük $\underline{z} = \underline{z}(Y)$ -al a $w(\underline{Y})$ -hoz tartozó $\underline{z}_k(\underline{Y})$ értékét. Ekkor az ÁLP-ben a $\underline{z}(\underline{Y})$ függvény az $\underline{Y}'$ pontban a $w(\underline{Y})$ függvény szubgradiense.

Bizonyítás: Mivel $w' = \max_{Y \in Q} w(\underline{Y})$, ezért

$$\begin{aligned} 0 &\leq w(\underline{Y}') - w(\underline{Y}) = \min_k (c_k + \underline{Y}' \underline{z}_k(\underline{Y}')) - \min_k (c_k + \underline{Y} \underline{z}_k(\underline{Y})) = \\ &= \min_k c_k + \underline{Y}' \min_k \underline{z}_k(\underline{Y}') - \min_k c_k - \underline{Y} \min_k \underline{z}_k(\underline{Y}) = \\ &= \underline{Y}' \underline{z}(\underline{Y}') - \underline{Y} \underline{z}(\underline{Y}) \leq (\underline{Y}' - \underline{Y}) \underline{z}(\underline{Y}) \end{aligned}$$

Mivel $0 \leq \underline{z}(\underline{Y})(\underline{Y}') - \underline{Y}$, ez azt jelenti, hogy az $(\underline{Y}' - \underline{Y})$ vektor és a $\underline{z}(\underline{Y})$ vektor hegyes szöget zárnak be. Legyen $\underline{Y}_0 \in Q$ egy kiindulási vektor. Legyen t_j egy olyan sorozat, amelyre

$$t_j > 0, \quad t_j \rightarrow 0, \quad \sum_{j=0}^{\infty} t_j = \infty \quad **$$

Bebizonyítható, hogy a t_j -k fenti választása mellett az

$$\underline{Y}_j = \underline{Y}_{j-1} + t_{j-1} \underline{z}(\underline{Y}_{j-1})$$

vektorsorozathoz tartozó

$$\underline{z}(\underline{Y}_j)(\underline{Y}' - \underline{Y}_j)$$

skalárszorzat 0-hoz konvergál. Ez azt jelenti, hogy

$$w(\underline{Y}') - w(\underline{Y}) \rightarrow 0$$

azaz a z szubgradiensek sorozatának a segítségével a $w(\underline{Y})$ függvény maximuma megtalálható. A szubgradiens optimalizáció algoritmusá tehát a következő:

1. Adjunk meg egy $\underline{Y}_0 \in Q$ -t és egy olyan t_j sorozatot, amely a ** feltételt kielégíti. $j=0$. Ugorjunk 3-ra.

2. $j = j+1$; $\underline{Y}_j = \underline{Y}_{j-1} + t_{j-1} \underline{z}(\underline{Y}_{j-1})$

3. Oldjuk meg a

$$w(\underline{Y}_j) = \min_k (c_k + \underline{Y}_j \cdot \underline{z}_k(\underline{Y}_j))$$

szélsőérték feladatot, ezáltal meghatározva a $\underline{z}(\underline{Y}_j)$ értékét.

4. Vizsgáljuk meg, hogy elég közel vagyunk-e az optimális megoldáshoz. Ha nem, ugorjunk 2-re, különben készen vagyunk.

/Egy lehetséges kritérium:

$$|w(\underline{Y}_j) - w(\underline{Y}_{j-1})| < \varepsilon$$

ahol $\varepsilon > 0$ egy előre adott pontosság./



A kérdés már csak az, hogy hogyan lehet megoldani a 3. pontban található minimum feladatot ?

Általános esetben a megoldással HELD és WOLFE [17] foglalkozott, a selejtminimalizálási feladatra B.L.GOLDEN [13] a következő megoldást javasolta:

Először nézzük meg, hogy milyen alaku lesz a selejtminimalizálási feladat, ha az ÁLP formára hozzuk. Induljunk ki az LP feladatból:

$$\sum_{j=1}^m a_{ij} x_j \geq n_i \quad i = 1, \dots, n$$

$$\sum_{j=1}^m x_j \leq \sum_{i=1} n_i \quad \text{xxx}$$

$$x_j \geq 0$$

$$\sum_{j=1}^m c_j x_j \rightarrow \min$$

ahol xxx feltétel egy redundáns egyenlőtlenség, amely az optimális szabástermben szereplő szabások számát korlátozza.

Ha az $\{\underline{X} \mid \sum x_j \leq \sum n_i, \underline{X} \geq 0\}$ halmaznak vannak extrémális pontjai, akkor a fenti LP a következő alakra hozható:

$$\sum_{k=1} (AX^k) \lambda_k \geq n$$

$$\sum_{k=1} \lambda_k = 1$$

$$\lambda_k \geq 0$$

$$\sum_{k=1} (CX^k) \lambda_k \rightarrow \min$$

A feladathoz rendelt duális

$$\begin{array}{rcl} \underline{Y} (\underline{AX}^k) + \underline{R} & \leq & \underline{CX}^k \quad k = 1, \dots, n \\ \underline{Y} & \geq & 0 \\ \hline \underline{Y} \underline{n} + \underline{R} & \rightarrow & \max \end{array}$$

Ez viszont az 1.4.1 TÉTEL-ben bebizonyítottakhoz hasonlóan megegyezik a

$$w(\underline{Y}) \rightarrow \max$$

feltéve
$$w(\underline{Y}) = \min_k \{ (\underline{C} - \underline{YA}) \underline{X}^k + \underline{Y} \underline{n} \}$$

feladattal.

Próbáljuk meg ez utóbbi kifejezést olyan alakra hozni, hogy jól látható legyen, hogy ez már ÁLP forma:

$$\begin{aligned} (\underline{C} - \underline{YA}) \underline{X}^k + \underline{Y} \underline{n} &= \sum_j c_j x_j^k - \sum_j x_j^k \sum_i y_i a_{ij} + \sum_i y_i n_i = \\ &= \sum_j c_j x_j^k - \sum_i \sum_j x_j^k y_i a_{ij} + \sum_i y_i n_i = \\ &= \sum_j c_j x_j^k - \sum_i y_i \sum_j x_j^k a_{ij} + \sum_i y_i n_i = \\ &= \sum_j c_j x_j^k + \sum_i y_i (n_i - \sum_j x_j^k a_{ij}) = \\ &= \sum_j \bar{c}_j + \sum_i y_i z_i = \bar{\underline{C}} + \underline{Y} \underline{z} \end{aligned}$$

A $z_i = n_i - \sum_j x_j^k a_{ij}$ -nek szemléletes jelentést is tulajdoníthatunk: z_i jelenti azt, hogy egy adott $w(\underline{Y})$ megoldás esetén az i . megrendelésből hány darabot nem gyártottunk le.

Az ÁLP feladat megoldásához induljunk ki a

$$x_j^k \min_{\geq 0} \{ \sum_j (c_j - \sum_i y_i a_{ij}) x_j^k + \sum_i y_i n_i \}$$

feltéve, hogy

$$\sum_j x_j^k = \sum_i n_i$$

Itt $\underline{a}_i = \{a_{ij}\}$ -t keressük /ami éppen egy szabás/. Keressük meg a legnegatívabb redukáló költséget:

$$\bar{c}_j = \min_j (c_j - \sum_i y_i a_{ij})$$

mivel $\sum_{i=1}^n y_i n_i$ adott

Ezt úgy tudjuk meghatározni, hogy megoldjuk a

$$\max_j \{ \sum_i y_i a_{ij} \mid \sum_i a_{ij} l_i \leq L \}$$

hátizsákfeladatot, vagy kiterjesztett esetben visszavezetjük a megoldást a fenti hátizsákfeladatra.

Ismereteink szerint a módszer hatékonyságára sem kísérleti, sem elméleti eredmények nem állnak rendelkezésre.

2. A HEURISZTIKÁRÓL ÁLTALÁBAN

A pillanatnyi eredmények azt mutatják, hogy számos kombinatorikus problémához nem található olyan módszer, amely pontos megoldást szolgáltat olyan idő alatt, amely az input adatok polinomiális függvényével meghatározható. /Az ilyen feladatot NP-teljesnek nevezzük./ Először HELD és Karp [16] bizonyította be, hogy a szabási probléma és a ládapakolási feladat is az NP teljes feladatok közé tartozik. Az ilyen feladatok megoldására gyakran alkalmaznak heurisztikus módszereket abban a reményben, hogy elfogadható időn belül elég pontos megoldást szolgáltatnak. Azt, hogy egy heurisztikus algoritmus mikor jó, nagyon nehéz eldönteni. Algoritmusokat legfeljebb úgy lehet összehasonlítani, hogy olyan méretű feladatokat oldunk meg, amelyeknek ismerjük a tényleges optimumát. Általában két módszert használnak az algoritmusok "jóságának" meghatározására:

- az átlagos eltérések

- a legrosszabb eset

vizsgálatát. Az előbbi módszer becsléseit valószínűségszámítás segítségével végzi úgy, hogy az alaphalmazra tesz valamilyen feltevést /pl. egyenletes eloszlás/, míg az utóbbi módszer kombinatorikus és számelméleti megfontolások segítségével ad felső korlátot. A legrosszabb eset vizsgálata előnyösebbnek tűnik több szempontból:

- az átlagos eset vizsgálatnál nehéz ellenőrizni, hogy a megrendelések halmaza eleget tesz-e a feltevéseknek.

- a legrosszabb eset vizsgálatának eredménye biztosabban kezelhető. Garantálja, hogy az ismert korlátnál rosszabb eredményt nem kapunk egy konkrét vizsgálatnál.
- A legrosszabb eset viselkedése általában hasonló az átlagos viselkedéshez.

A fentiek miatt viszonylag kevesebb olyan eredmény van, amely az átlagos eltérések vizsgálatával foglalkozik. Mi sem foglalkozunk részletesebben az átlagos eltérések módszerével. Itt mindössze annyit említünk meg, hogy S.D. SHAPIRO [29] és G.N.FREDRICKSON [6] vizsgált részletesebben heurisztikákat egy- illetve kétdimenziós esetben.

Dolgozatunknak ebben a fejezetében azokat az eredményeket mutatjuk be, amelyeket a legrosszabb eset vizsgálatakor kaptak a heurisztikák általános elemzésekor. Ismertetjük azokat az eredményeket, amelyeket JOHNSON és munkatársai [21] kaptak néhány speciális algoritmus vizsgálatakor. A fejezet utolsó részében bemutatunk egy egyszerű kétdimenziós eljárást és elvégezzük ennek legrosszabb eset vizsgálatát.

2.1. Ládapakolási heurisztikák egydimenzióban

Alapfeladat: Legyen adva a valós számok egy $L = (a_1, \dots, a_n)$ vektora, ahol $a_i \in (0, 1]$. Helyezzük el L elemeit L^* minimális számúládában úgy, hogy az egy ládában elhelyezett számok összege ne legyen nagyobb 1-nél.

Ilyen típusú feladatok heurisztikus algoritmus segítségével történő megoldására már CHRISTOFIDES [5] is közölt algoritmust, ő azonban nem foglalkozott részletesebben a heurisztika analízisével. Az első jelentős analízist JOHNSON [20] végezte el az általa ismertetett algoritmusokon. Később munkatársaival négy heurisztikát vizsgált meg a regresszabb eset analízisének segítségével [21]. A legismertebb heurisztikák ismertetése előtt vizsgáljuk meg azokat az eredményeket, amelyeket az egydimenziós heurisztikák általános vizsgálatával kaphatunk.

Definíció: Legyen $E = \{a \mid 0 < a \leq 1\}$. Legyen továbbá

$N = \{1, \dots, n\}$, és $v: N \rightarrow E$. Legyen $\mathcal{L} = \{S_1, S_2, \dots, S_m\}$ N -nek egy osztályozása. Azt mondjuk, hogy $\mathcal{L}$ v-megengedett, ha $\forall_j \in M$ -re

$$\sum_{i \in S_j} v(i) \in E$$

Megjegyzés: A definícióból látható, hogy v az E halmaz egy részhalmazát szolgáltatja, és egy v -megengedett osztályozás megfelel egy ládapakolásnak.

Definíció: Egy olyan f leképezését, amely bármely L listához egy $\mathcal{L}$ v -megengedett osztályozást hoz létre, a heurisztika szabályának nevezzük.

Mivel a heurisztikus algoritmusokra gyakorlatilag semmilyen megkötésünk nincs, célszerűnek látszik definiálni azt, hogy melyek azok az algoritmusok, amelyek elfogadható eredményt szolgáltatnak. Egy kritérium lehet az ésszerűsége, ha azt követeljük meg a heurisztikától, hogy az általa létrehozott osztályozás irreducibilis legyen:

Definíció: Legyen $\mathcal{L}^k = \{S_l^k \mid S_l^k = \bigcup_{j=1}^k S_{i_j}, i_j \in M\}$. Az f heurisztika által létrehozott $\mathcal{L}_f(L)$ osztályozást k -irreducibilisnek nevezzük, ha $\mathcal{L}_f(L)$ v -megengedett és semelyik $S_l^k \in \mathcal{L}^k$ -nak nincs olyan v -megengedett $\mathcal{L}'_f, (S_l^k)$ osztályozása amelyre $|\mathcal{L}'_f| \leq k-1$

Megjegyzés: A fenti definíció azt jelenti, hogy bárhogyan is választunk ki k ládát az $\mathcal{L}_f(L)$ osztályozásból, azok tartalmát semmiképpen nem tudjuk $k-1$ ládában elhelyezni.

2.1.1. TÉTEL: Ha egy f heurisztika által létrehozott $\mathcal{L}_f(L)$ osztályozás nem k -irreducibilis, akkor minden k -nál magasabb l szinten is nem l -irreducibilis.

Bizonyítás: Ha $\mathcal{L}_f(L)$ nem k -irreducibilis, akkor kiválasztható belőle k darab olyan láda, amelynek tartalma elhelyezhető $k-1$ ládában.

Minden l magasabb szinthez válasszuk ki ezt a k ládát és ezeket egészítsük ki l -re. Világos, hogy ezen l db ládának a tartalma mindig elhelyezhető $l-1$ ládában, tehát az osztályozás nem l -irreducibilis.

Legyen $v: \mathbb{N} \rightarrow E$ a következő L halmaz: $L = (a_1, \dots, a_n)$ LCE. Tegyük fel, hogy L -t optimálisan L^* "ládába" tudjuk elhelyezni. Ekkor egy f heurisztika hatékonyságának mérésére az

$$R_f(L) = \frac{|\mathcal{L}_f(L)|}{L^*}$$

hányadost használjuk, ahol $|\mathcal{L}_f(L)|$ jelöli annak az osztályozásnak a számosságát, amelyet az f heurisztikával az L halmazból állítottunk elő.

Definíció: Legyen $\mathcal{F}$ az olyan f heurisztikák halmaza, amelyek v -megengedett 2-irreducibilis osztályozást hoznak létre. Legyen

$$\zeta(m) = \min \left\{ \sum_{i=1}^m f(i) \mid f \in \mathcal{F} \right\}$$

$$\text{ahol } m = |\mathcal{L}_f(L)|$$

2.1.2. TÉTEL: Minden v -megengedett 2-irreducibilis osztályozást létrehozó heurisztikára

$$m \geq \zeta(m)$$

Bizonyítás:

$$m \geq L^* \geq \sum_{i=1}^m v(i) = \sum_{i=1}^m \sum_{k \in S_i} v(k) = \sum_{i=1}^m f(i) \geq \zeta(m)$$

Igy igaz, hogy

$$R_f(L) = \frac{|\mathcal{L}_f(L)|}{L^*} \leq \frac{|\mathcal{L}_f(L)|}{\zeta} \leq \frac{m}{\lceil \zeta(m) \rceil}$$

ahol $\lceil \rceil$ jelenti a számnál nagyobb vagy egyenlő egészek közül a legkisebbet.

KOU és MARKOWSKY bizonyította be a következő tételt [25]:

2.1.3. TÉTEL: Minden $m \geq 2$ -re $\zeta(m) = \lfloor m/2 \rfloor + \lambda(m)$, ahol

$$\lambda(m) = \begin{cases} \phi & , \text{ ha } m \text{ páros} \\ \frac{1}{2} & , \text{ ha } m \text{ páratlan} \end{cases}$$

A fenti tétel felhasználásával bebizonyítható a következő

2.1.4. TÉTEL: Legyen $f \in \mathcal{F}$. Ekkor

$$R_f(L) \leq 2$$

Bizonyítás:

$$R_f(L) = \frac{|\mathcal{L}_f(L)|}{L^*} \leq \frac{m}{\lceil \zeta(m) \rceil} = \frac{m}{\left\lceil \left\lfloor \frac{m}{2} \right\rfloor + \lambda(m) \right\rceil} = \frac{m}{\left\lceil \frac{m}{2} \right\rceil} \leq 2$$

A tétel azt jelenti, hogy elfogadhatónak tekintjük azokat a legalább 2-irreducibilis osztályozást nyújtó f heurisztikákat, amelyekre $|\mathcal{L}_f(L)| / L^* \leq 2$, azaz legfeljebb kétszer annyi ládát használnak el, mint amennyi az optimumhoz szükséges.

Megjegyzés: KOU és MARKOWSKY a fenti tételeket bebizonyította k dimenzióra. A tételt itt bizonyítás nélkül közöljük; a bizonyítás megtalálható [25]-ben.

2.1.5. TÉTEL: Legyen $\theta^k = \{(a_1, \dots, a_k) \mid 0 < a_i \leq 1\}$. Legyen $v: \mathbb{N} \rightarrow \theta^k$. Legyen f egy olyan heurisztika, amely v -megengedett 2-irreducibilis osztályozást hoz létre. Ekkor

$$a/ \quad \frac{|\mathcal{L}_f(L)|}{L^*} \leq k - 1$$

b/ minden $k \geq 1$ -re és $\delta > 0$ -ra van olyan f , amelyre

$$\frac{|\mathcal{L}_f(L)|}{L^*} \geq k - 1 - \delta$$

Ismertebb heurisztikák egydimenzióban

A következőkben négy olyan heurisztikus algoritmust ismertetünk, amelynek legrosszabb eset vizsgálatát először JOHNSON végezte el. Megmutatjuk, hogy a JOHNSON-féle alsó korlát javítható, és bemutatjuk azt a tételt, amelyet JOHNSON és munkatársai [21] ismertettek a heurisztikák R hányadosának aszimptotikus határértékére $L^* \rightarrow \infty$ esetben.

A vizsgált négy módszer

a./ Első illesztés /First Fit: FF/

Tekintsük az alapfeladatot. Legyen $v: \mathbb{N} \rightarrow E$ eredménye $L = (a_1, \dots, a_n)$ halmaz. Legyen az $f=FF$ heurisztika a következő: Álljon rendelkezésre $B_1, B_2, \dots$ végtelen számú láda, amelyek kezdetben legyenek $\beta_i = 0$ szinten töltve. Amikor az a_i számot próbáljuk elhelyezni, akkor azt a legkisebb j -t keressük, amelyre $B_j \beta_j$ szinten van töltve és $\beta_j \leq 1 - a_i$. Ebben helyezzük el a számot. Legyen tehát

$$J = \min_j \{B_j \mid \beta_j \leq 1 - a_i\}$$



Az elhelyezés után $\beta'_j = \beta_j + a_i$.

Ekkor $S_i \in_{FF}(L)$ -be azok a számok tartoznak, amelyek a B_i ládába lettek elhelyezve.

b./ Legjobb illesztés /Best Fit: BF/

Tekintsük az alapfeladatot. Legyen $v: N \rightarrow E$ eredménye az $L = (a_1, \dots, a_n)$ halmaz. Legyen az $f = BF$ heurisztika a következő: Álljon rendelkezésre $B_1, B_2, \dots$ végtelen számu láda, amelyek kezdetben $\beta_i = 0$ szinten vannak töltve. Amikor a_i -t akarjuk elhelyezni, akkor keressük meg azt a B_j -t, amelyhez tartozó β_j -re igaz, hogy $\beta_j \leq 1 - a_i$, és β_j a lehető legnagyobb. Tehát

$$J = \min_j \{ \max_{\beta_j} B_j \mid \beta_j \geq 1 - a_i \}$$

Az elhelyezés után $\beta'_j = \beta_j + a_i$

Ekkor $S_i \in_{BF}(L)$ -be azok a számok tartoznak, amelyek a B_j ládába lettek elhelyezve.

c./ Első illesztés rendezéssel /First Fit Decreasing: FFD/

Tekintsük az alapfeladatot. Legyen $v: N \rightarrow E$ eredménye egy $L = (a_1, \dots, a_n)$ halmaz. Legyen $f = FFD$ heurisztikánk a következő: Rendezzük az L halmaz elemeit úgy, hogy $a_i \geq a_j$ álljon, ha $i < j$, majd az így kapott halmaz elemeire alkalmazzuk az FF heurisztikát. Az FFD heurisztikához tartozó osztályozást jelöljük $\mathcal{L}_{FFD}(L)$ -el.

d./ Legjobb illesztés rendezéssel /Best Fit Decreasing:

BFD/

Tekintsük az alapfeladatot. Legyen $v: \mathbb{N} \rightarrow E$ eredménye egy $L=(a_1, \dots, a_n)$ halmaz. Legyen az $f=BFD$ heurisztikánk a következő: Rendezzük az L halmaz elemeit úgy, hogy $a_i \geq a_j$ álljon, ha $i < j$, majd az így kapott halmaz elemeire alkalmazzuk a BF heurisztikát. A BFD heurisztikához tartozó $\mathcal{L}$ osztályozást jelöljük $\mathcal{L}_{BFD}(L)$ -el.

2.1.6. TÉTEL: Az FF heurisztika által létrehozott $\mathcal{L}_{FF}(L)$ osztályozás bármely L halmazra 2-irreducibilis.

Bizonyítás: Legyen az $\mathcal{L}_{FF} L = \{S_1, \dots, S_m\}$ és legyen $S_i \beta_i, S_j \beta_j$ szintig töltve a heurisztika befejezésekor. Megmutatjuk, hogy $\forall i, j$ -re $\beta_i + \beta_j > 1$. Legyen az $S_j \beta'_j$ szinten töltve mielőtt az utolsó a_{k_j} -elemet elhelyeztük benne. Az általánosság megszorítása nélkül feltehetjük, hogy $i < j$. Ekkor

$$1 - \beta_i < a_{k_j} \leq 1 - \beta'_j$$

innen

$$1 < a_{k_j} + \beta_i$$

Legyen

$$\beta_j = a_{k_j} + \beta'_j$$

Ekkor

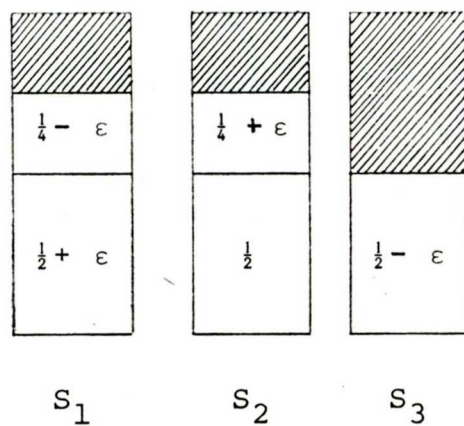
$$\beta_i + \beta_j = \beta'_j + a_{k_j} + \beta_i > \beta'_j + 1 > 1$$

Megjegyzés: kimutatható, hogy az FF heurisztika már nem 3-irreducibilis. Elegendő megadni egy olyan L halmazt, amelyre nem teljesül a feltétel. Pl.

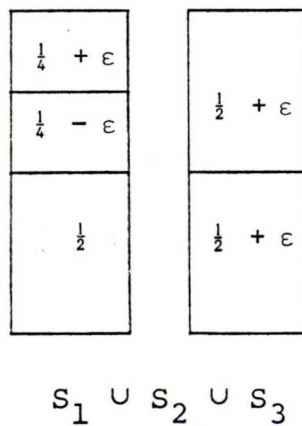
$$L = \frac{1}{2} + \varepsilon ; \frac{1}{4} - \varepsilon ; \frac{1}{2} ; \frac{1}{4} + \varepsilon ; \frac{1}{2} - \varepsilon . \text{ Ahol } \varepsilon > \frac{1}{12}$$

Ekkor

$$\mathcal{L}_{FF}(L) = \{S_1, S_2, S_3\}$$



Itt az $S_1 \cup S_2 \cup S_3$ elhelyezhető két ládában.



2.1.7. TÉTEL: A BF heurisztikával létrehozott $\mathcal{L}_{BF}(L)$ osztályozás bármely L halmazra 2-irreducibilis.

Bizonyítás: Tegyük fel, hogy egy adott állapotban az a_{k_j} elemet utolsóként helyeztük a B_j ládába. Legyen a B_i láda β_i szintig, a B_j láda β_j szintig töltve az elhelyezés előtt. Ekkor

$$1 - \beta_j' \geq a_{k_j}$$

és

$$\beta_j' = \max_{\beta} \{ \beta \mid 1 - \beta \geq a_{k_j} \}$$

Legyen

$$\beta_j = \beta_j' + a_{k_j}$$

a/ Tegyük fel, hogy $1 - \beta_i < a_{k_j}$. Ekkor az FF heurisztikánál bizonyítottak szerint igaz, hogy $\beta_i + \beta_j > 1$.

b/ Legyen $1 - \beta_i \geq a_{k_j}$. Ekkor kellett lenni egy olyan előző állapotnak, amikor $\beta_i' > \beta_j'$ állt és B_j -be olyan elemet tettünk - a_{k_j}' - amely B_i -ben már nem volt elhelyezhető, azaz β_i' , β_j' és a_{k_j}' -re már teljesültek az a/ pont feltételei.

Amennyiben a_{k_j} első elem volt B_j -ben, akkor automatikusan igaz a tétel állítása, hiszen a BF algoritmus-sal új ládát csak akkor kezdünk, ha az elem már egyetlen megkezdett ládában sem fér el.

A fenti tételekhez hasonlóan belátható, hogy

2.1.8. TÉTEL: Az FFD (BFD) heurisztika által létrehozott $\mathcal{L}_{FFD}(L)$ ($\mathcal{L}_{BFD}(L)$) osztályozás 2-irreducibilis

Az előzőekből következik, hogy bármely L listára igaz, hogy

$$R_{FF}(L) = \frac{|\mathcal{L}_{FF}(L)|}{L^*} \leq 2 \quad R_{FFD}(L) = \frac{|\mathcal{L}_{FFD}(L)|}{L^*} \leq 2$$

$$R_{BF}(L) = \frac{|\mathcal{L}_{BF}(L)|}{L^*} \leq 2 \quad R_{BFD}(L) = \frac{|\mathcal{L}_{BFD}(L)|}{L^*} \leq 2$$

JOHNSON és munkatársai [21]-ben részletesen vizsgálva a fenti heurisztikákat, ennél élesebb eredményeket kaptak. Bebizonyították, hogy

2.1.9. TÉTEL Bármely $k > 0$ egész számra megadható egy olyan L lista, amelyre $L^* = k$ és

$$R_{FF}(L) = \frac{|\mathcal{L}_{FF}(L)|}{L^*} \geq \frac{17}{10} - \frac{8}{L^*}$$

A bizonyításhoz egy viszonylag bonyolult L listát adtak meg. Ennek részletes ismertetésétől itt eltekintünk, mert hosszadalmas. A bizonyítás két részből áll. Az első részben azokkal a listákkal foglalkoznak, amelyekre $|\mathcal{L}_{FF}(L)| = k \equiv 1 \pmod{10}$. Ekkor a lista olyan, hogy optimális esetben $L^* = 10N/17+1$ ládában helyezhető el. Ugyanerre a listára alkalmazva az FF heurisztikát, azt kapták, hogy csak olyan osztályozás adható meg, amelyre

$$|\mathcal{L}_{FF}(L)| = N$$

Igy azt kapták, hogy

$$\begin{aligned} R_{FF}(L) &= \frac{|\mathcal{L}_{FF}(L)|}{L^*} = \frac{17N}{10N + 17} = \frac{170N}{10(17+10N)} = \frac{17}{10} - \frac{\frac{289}{10}}{17+10N} = \\ &= \frac{17}{10} - \frac{\frac{17}{10}}{1 + \frac{10}{17N}} \geq \frac{17}{10} - \frac{17}{L^*} \geq \frac{17}{10} - \frac{2}{L^*} \end{aligned}$$

JOHNSON úgy lépett tovább, hogy abban az esetben, amikor $|\mathcal{L}_{FF}(L)| = k \not\equiv 1 \pmod{10}$, akkor egységnyi elemekkel egészítette ki a korábbiakban alkalmazott listát. Ekkor

$$L' = L + m \quad (m \leq 9),$$

$$|\mathcal{L}_{FF}(L')| = |\mathcal{L}_{FF}(L)| + m = N + m \text{ és}$$

$$L'^* = L^* + m = \frac{10N}{17} + m + 1$$

Igy

$$\begin{aligned} \frac{|\mathcal{L}_{FF}(L')|}{L'^*} &= \frac{N+m}{L^*+m} \geq \frac{10N+10m}{10(\frac{10N}{17}+m+1)} = \frac{17}{10} - \frac{7m+17}{10(\frac{10N}{17}+m+1)} \geq \\ &\geq \frac{17}{10} - \frac{7m+17}{10 L^*} \geq \frac{17}{10} - \frac{8}{L^*} \end{aligned}$$

JOHNSON fenti tételét meg lehet javítani úgy, hogy a bizonyítás második felében az eddig felhasznált m egységnyi szám helyett ismét egy megfelelően konstruált listát szerkesztünk és ezt összefésüljük az eredeti listával. A javítással nyerhető tétel a következő:

2.1.10. TÉTEL: Bármely $k > 0$ egész számhoz található olyan L lista, amelyre $L^* = k$ és

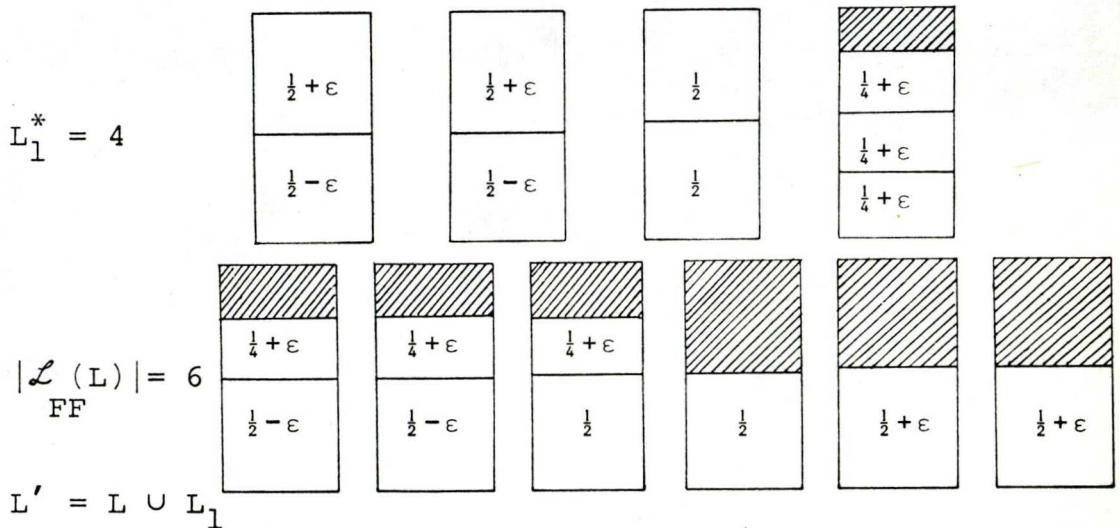
$$R_{FF}(L) = \frac{|\mathcal{L}_{FF}(L)|}{L^*} \geq \frac{17}{10} - \frac{4}{L^*}.$$

Bizonyítás: Legyen $\bar{k} \equiv k(\text{mod } 10)$, ahol $0 \leq \bar{k} \leq 9$. A tétel állítása azonnal következik a Johnson-tételből, ha $\bar{k} \leq 3$. Ugyanis annak kell teljesülni, hogy

$$\frac{7m + 17}{10} \leq 4 \Rightarrow m \leq 3$$

a/ Legyen $\bar{k}=4$

$$L_1: \frac{1}{2} - \varepsilon; \frac{1}{4} + \varepsilon; \frac{1}{2} - \varepsilon; \frac{1}{4} + \varepsilon; \frac{1}{2}; \frac{1}{4} + \varepsilon; \frac{1}{2} + \frac{1}{2} + \varepsilon; \frac{1}{2} + \varepsilon$$



$$R_{FF}(L') = \frac{|\mathcal{L}_{FF}(L')|}{L'^*} = \frac{N+6}{L^*+4} \geq \frac{10N+60}{10(\frac{10N}{17}+5)} \geq \frac{17}{10} - \frac{25}{10L'^*} \geq \frac{17}{10} - \frac{3}{L'^*}$$

b/ Legyen $\bar{k} = 5$

$$L_1: \frac{1}{2} - \varepsilon; \frac{1}{4} + \varepsilon; \frac{1}{2} - \varepsilon; \frac{1}{4} + \varepsilon; \frac{1}{2} - \varepsilon; \frac{1}{4} + \varepsilon; \frac{1}{2}; \frac{1}{2} + \varepsilon; \frac{1}{2} + \varepsilon; \frac{1}{2} + \varepsilon$$

$$L_1^* = 5$$

$\frac{1}{2} - \epsilon$	$\frac{1}{2} - \epsilon$	$\frac{1}{2} - \epsilon$	$\frac{1}{4} + \epsilon$	
$\frac{1}{2} + \epsilon$	$\frac{1}{2} + \epsilon$	$\frac{1}{2} + \epsilon$	$\frac{1}{2}$	$\frac{1}{4} + \epsilon$

$$|\mathcal{L}_{FF}(L_1)| = 7$$

$\frac{1}{4} + \epsilon$	$\frac{1}{4} + \epsilon$	$\frac{1}{4} + \epsilon$				
$\frac{1}{2} - \epsilon$	$\frac{1}{2} - \epsilon$	$\frac{1}{2} - \epsilon$	$\frac{1}{2}$	$\frac{1}{2} + \epsilon$	$\frac{1}{2} + \epsilon$	$\frac{1}{2} + \epsilon$

$$R_{FF}(L') = \frac{|\mathcal{L}_{FF}(L')|}{L'^*} = \frac{N+7}{L^*+5} \geq \frac{10N+70}{10(\frac{10N}{17}+6)} \geq \frac{17}{10} - \frac{32}{10L'^*} \geq \frac{17}{10} - \frac{4}{L'^*}$$

c/ Legyen $\bar{k} = 6$

$$L_1: \frac{1}{2}; \frac{1}{4} + \epsilon; \frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon; \frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon; \frac{1}{2} - \epsilon; \frac{1}{4} - \epsilon; \frac{1}{2};$$

$$\frac{1}{2} + \epsilon; \frac{1}{2} + \epsilon; \frac{1}{2} + \epsilon; \frac{1}{2}$$

$$L_1^* = 6$$

$\frac{1}{2}$	$\frac{1}{2} - \epsilon$	$\frac{1}{4} + \epsilon$	
$\frac{1}{2}$	$\frac{1}{2} + \epsilon$	$\frac{1}{2}$	$\frac{1}{4} + \epsilon$

3x

$$|\mathcal{L}_{FF}(L_1)| = 9$$

$\frac{1}{4} + \epsilon$	$\frac{1}{4} + \epsilon$			
$\frac{1}{2}$	$\frac{1}{2} - \epsilon$	$\frac{1}{2}$	$\frac{1}{2} + \epsilon$	$\frac{1}{2}$

3x

3x

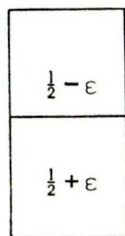
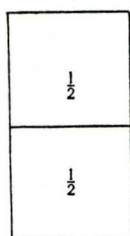
$$R_{FF}(L') = \frac{|\mathcal{L}_{FF}(L')|}{L'^*} = \frac{N+9}{L^*+6} \geq \frac{10N+90}{10(\frac{10N}{17}+7)} \geq \frac{17}{10} - \frac{29}{10L'^*} \geq \frac{17}{10} - \frac{3}{L'^*}$$

d/ Legyen $\bar{k} = 7$

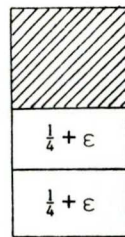
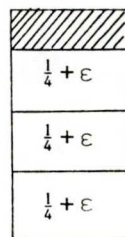
$$L_1: \frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon; \frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon; \frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon; \frac{1}{2}; \frac{1}{4} + \epsilon; \frac{1}{2};$$

$$\frac{1}{2} + \epsilon; \frac{1}{2} + \epsilon; \frac{1}{2} + \epsilon; \frac{1}{2} + \epsilon$$

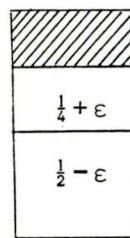
$$L_1^* = 7$$



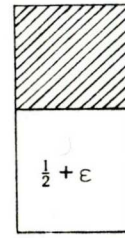
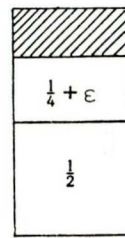
4x



$$|\mathcal{L}_{FF}(L_1)| = 10$$



4x



3x

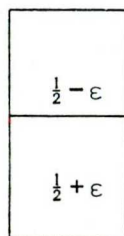
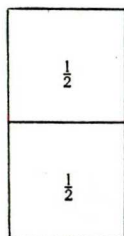
$$R_{FF}(L') = \frac{|\mathcal{L}_{FF}(L')|}{L'^*} = \frac{N+10}{L^*+7} \geq \frac{17}{10} - \frac{36}{10L'^*} \geq \frac{17}{10} - \frac{4}{L'^*}$$

e/ Legyen $\bar{k} = 8$

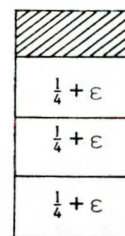
$$L_1: \frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon; \frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon; \frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon; \frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon;$$

$$\frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon; \frac{1}{2}; \frac{1}{4} + \epsilon; \frac{1}{2}; \frac{1}{2} + \epsilon; \frac{1}{2} + \epsilon; \frac{1}{2} + \epsilon; \frac{1}{2} + \epsilon; \frac{1}{2}$$

$$L_1^* = 8$$

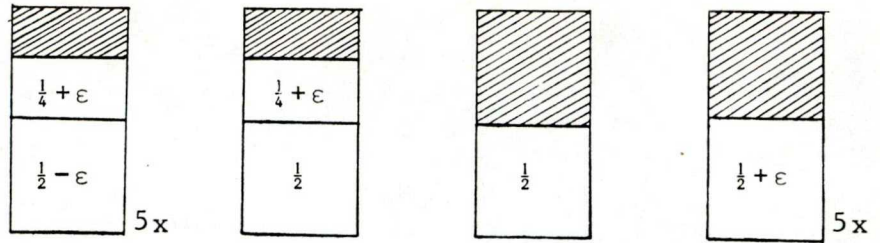


5x



2x

$$|\mathcal{L}_{FF}(L_1)| = 12$$



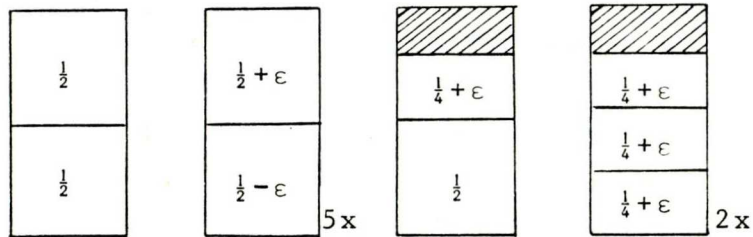
$$R_{FF}(L') = \frac{|\mathcal{L}_{FF}(L')|}{L'^*} = \frac{N+12}{L'^*+8} \geq \frac{17}{10} - \frac{33}{10L'^*} \geq \frac{17}{10} - \frac{4}{L'^*}$$

f/ Legyen $\bar{k} = 9$

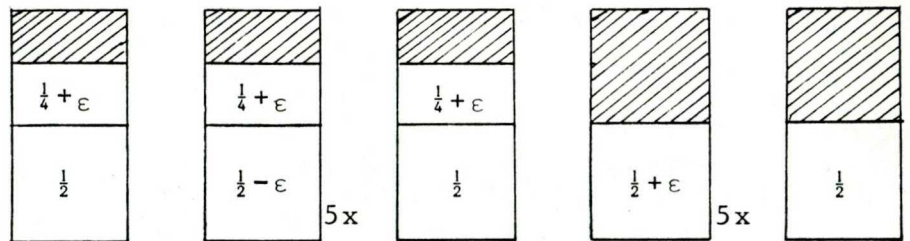
$$L_1: \frac{1}{2}; \frac{1}{4} + \epsilon; \frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon; \frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon; \frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon; \frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon;$$

$$\frac{1}{2} - \epsilon; \frac{1}{4} + \epsilon; \frac{1}{2}; \frac{1}{4} + \epsilon; \frac{1}{2} + \epsilon; \frac{1}{2} + \epsilon; \frac{1}{2} + \epsilon; \frac{1}{2} + \epsilon; \frac{1}{2} + \epsilon; \frac{1}{2}$$

$$L_1^* = 9$$



$$|\mathcal{L}_{FF}(L_1)| = 13$$



$$R_{FF}(L') = \frac{|\mathcal{L}_{FF}(L')|}{L'^*} = \frac{N+13}{L'^*+9} \geq \frac{17}{10} - \frac{40}{10L'^*} = \frac{17}{10} - \frac{4}{L'^*}$$

Ez azt jelenti, hogy bármely $\bar{k} \leq 9$ -re igaz, ha $k=10N+\bar{k}$, akkor mindig adható olyan L lista, amelyre $L^* = k$ és

$$R_{FF}(L) \geq \frac{17}{10} - \frac{4}{L^*}$$

JOHNSON alsó korlátot talált az FFD heurisztikára is [20]:

2.1.11. TÉTEL: Legyen $k > 0$ természetes szám. Ekkor mindig megadható olyan L lista, amelyre $L^* = k$ és

$$R_{FFD}(L) = \frac{|\mathcal{L}_{FFD}(L)|}{L^*} \leq \frac{11}{9} - \frac{2}{L^*}$$

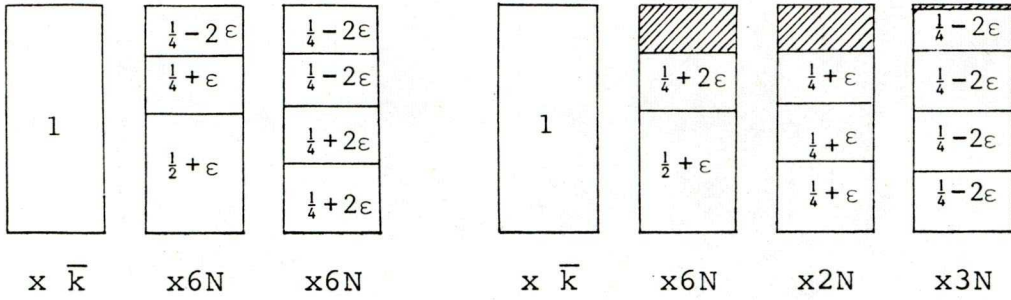
Bizonyítás: A bizonyítás abból áll, hogy megkonstruálunk egy ilyen listát.

Legyen $N = \left\lceil \frac{k}{9} \right\rceil$, $\bar{k} \equiv k \pmod{9}$, $0 \leq \bar{k} \leq 9$,

$n = 30N + \bar{k}$, $L = (a_1, \dots, a_n)$, ahol

$$a_i = \begin{array}{ll} \frac{1}{2} + \varepsilon & 1 \leq i \leq 6N \\ \frac{1}{4} + 2\varepsilon & 6N < i \leq 12N \\ \frac{1}{4} + \varepsilon & 12N < i \leq 18N \\ \frac{1}{4} - 2\varepsilon & 18N < i \leq 30N \\ 1 & 30N < i \leq n \end{array}$$

Erre a listára igaz, hogy $L^* = 9N + \bar{k}$ és $|\mathcal{L}_{FFD}(L)| = 11N + \bar{k}$



$$|\mathcal{L}_{\text{FFD}}(L)| = 11N + \bar{k} = L^* + 2N + \bar{k} = L^* + \frac{2L}{9} - \frac{2\bar{k}}{9} \geq \frac{11}{9} L^* - 2$$

2.1.12. TÉTEL: Legyen L egy tetszőleges lista. Ekkor

a/ $|\mathcal{L}_{\text{FF}}(L)| \leq \frac{17}{10} L^* + 2$

b/ $|\mathcal{L}_{\text{BF}}(L)| \leq \frac{17}{10} L^* + 2$

c/ $|\mathcal{L}_{\text{FFD}}(L)| \leq \frac{11}{9} L^* + 2$

d/ $|\mathcal{L}_{\text{BFD}}(L)| \leq \frac{11}{9} L^* + 2$

A fenti tételek bizonyítását nem ismertetjük, mivel hosszadalmas. A bizonyítás [21]-ben megtalálható. Ezeknek a tételeknek egyenes következménye a következő tétel:

2.1.13. TÉTEL: Legyen $R_{\text{FF}}(k) = \max \frac{|\mathcal{L}_{\text{FF}}(L)|}{L^*}$

ahol a maximumot azokra az L listákra számítjuk, amelyekre $L^* = k$. Ekkor

a/ $\lim_{k \rightarrow \infty} R_{\text{FF}}(k) = \lim_{k \rightarrow \infty} R_{\text{BF}}(k) = \frac{17}{10}$

b/ $\lim_{k \rightarrow \infty} R_{\text{FFD}}(k) = \lim_{k \rightarrow \infty} R_{\text{BFD}}(k) = \frac{11}{9}$

2.2. HEURISZTIKÁK KÉTDIMENZIÓBAN

Az egydimenzióra kapott eredményeket már JOHNSON is kiterjesztette magasabb dimenzióra. A kiterjesztéshez két lehetőség közül választhatunk:

- 1./ Olyan általánosítás, amely a többdimenziós korlátokkal történő rakodási feladat megoldására alkalmas /lásd 35. oldal/. A kiterjesztés ebben az esetben a következő definíción alapul:

Definíció: Legyen $k \geq 1$ és legyen $\theta^k = \{(a_1, \dots, a_k) \mid 0 \leq a_i \leq 1\}$.
Legyen továbbá $N = \{1, 2, \dots, n\}$ és $v: N \rightarrow \theta^k$. Azt mondjuk, hogy N -nek $\mathcal{L} = \{S_1, \dots, S_m\}$ osztályozása v -megengedett, ha $\forall n$ -re

$$\sum_{i \in S_j} (v)_i \in \theta^k$$

KOU és MARKOWSKY a fenti definícióra építve bizonyította be az előző részben ismerttetett tételeket.

- 2./ Ha azt a megfeleltetést akarjuk tovább vinni, amely az egydimenziós ládapakolási feladat és a szabási probléma között fennáll, akkor tekinthetjük a következő általánosítást:

Definíció: Legyen $k \geq 1$ és $\theta^k = \{(a_1, \dots, a_k) \mid 0 \leq a_i \leq 1\}$.
Legyen továbbá $N = \{1, 2, \dots, n\}$ és $v: N \rightarrow \theta^k$. Azt mondjuk, hogy N -nek $\mathcal{L} = \{S_1, \dots, S_m\}$ osztályozása egy k -dinemziós szabásterv, ha $\forall S_i \in \mathcal{L}$ -hez megadható egy olyan f_i leképezés, amely S_i -t a θ^k -beli téglalapokra képezi le, amelyre

$$1./ \quad \forall x \in S_i \Rightarrow f_i(x) \text{ "alakja" } v(x)$$

$$2./ \quad x \neq y \in S_i \Rightarrow \text{int } f_i(x) \cap \text{int } f_i(y) = \emptyset$$

Megjegyzés: A definíció közvetlen következménye, hogy k-dimenziós szabásterv esetén

$$\sum_{j \in S_i} \left(\prod_{i=1}^k a_i \mid v(j) = (a_1, \dots, a_k) \right) \leq 1$$

A kétdimenziós heurisztika megértését elősegítő definíciók az első fejezetben a Gilmore-Gomory módszer általánosításánál találhatók meg. Látható, hogy abban az esetben, amikor a megrendelések nem forgathatók, akkor a fenti ládapakolási problémának koordinátatranszformációval mindig megfeleltethető egy szabási feladat. Ilyen típusu kétdimenziós szabási feladatra közölt heurisztikát ALBANO és ORSINI [1], valamint FREDRICKSON [6] is. Ezekben a heurisztikákban az egyik fő kikötés az, hogy csak guillotine típusu vágásokat engednek meg. A dolgozatnak ebben a részében egy olyan heurisztikát vizsgálunk, amely egzakt kétütemű vágásokat enged meg olyan megrendeléseken, amelyek nem forgathatók.

Guillotine heurisztika /GH/ kétdimenzióban

Jelöljük a ládákban elhelyezendő tárgyak halmazát

$L: \{(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)\}$ -el. A heurisztika lényege az, hogy csikok képzésével visszavezetjük a feladatot egydimenziós ládapakolási feladatok sorozatára:

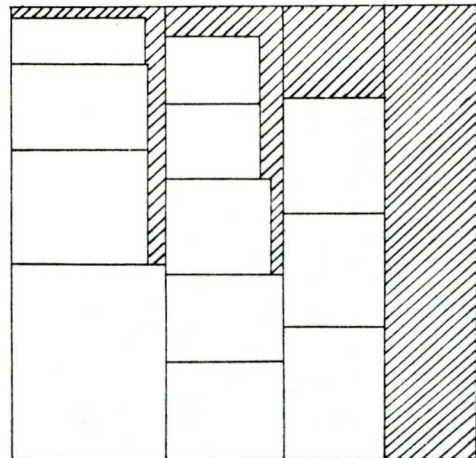
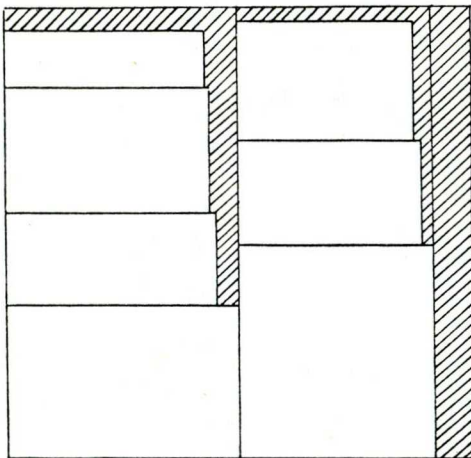
1. Rendezzük az L lista elemeit csökkenő sorrendbe az első koordináta szerint. Ekkor $a_i \geq a_j$, ha $i < j$

2. Vegyük ki az első elemet a listából. Jelöljük ki egy a_1 szélességű csikot /ez megfelel egy csikszabásnak/.
3. Helyezzük el a soron következő lista elemét az első olyan csikban, amelyben elfér. Ha nem helyezhető el egyetlen csikban sem, akkor az aktuális elem első koordinátájával hozzunk létre egy újabb csikot. Jelöljük a k . ládában létrehozott csikszélességeket $a_{k_1}, \dots, a_{k_n}$ -el, és legyen az aktuális listaelem az (a_j, b_j) .
4. Keressük meg azt a legkisebb indexű ládát, amelyre

$$1 - \sum_{i=1}^n a_{k_i} \geq a_j$$

és helyezzük el ebben a ládában az elemet.

5. Ha nincs ilyen B láda, akkor kezdünk új ládát. Folytassuk az eljárást a 3. ponttól, amíg el nem helyeztük az összes elemet.



Ládapakolási képek GH heurisztikával

Észrevehetjük, hogy tulajdonképpen az FF és BFD algoritmusokat használjuk fel. Kimondhatjuk a következő tételt:

2.2.1. TÉTEL: Legyen k tetszőleges szám. Ekkor mindig megadható olyan $L = \{(a_1, b_1), \dots, (a_n, b_n)\}$ $0 < a_i, b_i \leq 1$ kétdimenziós lista, amelyet optimális esetben $L^* = k$ ládában helyezhetünk el, míg a GH algoritmusra

$$|\mathcal{L}_{GH}(L)| \geq \frac{187}{90} L^* - 20$$

Bizonyítás: Legyen k tetszőleges és legyenek

$$N = \left\lfloor \frac{k}{90} \right\rfloor \quad \bar{k} \equiv k \pmod{90} \quad k = 90N + \bar{k} \quad 0 \leq \bar{k} < 90$$

$$\bar{N} = \left\lfloor \frac{\bar{k}}{9} \right\rfloor \quad \bar{\bar{k}} \equiv \bar{k} \pmod{9} \quad k = 90N + 9\bar{N} + \bar{\bar{k}} \quad 0 \leq \bar{\bar{k}} < 9$$

Legyen $n = 300N + 30\bar{N} + \bar{\bar{k}}$ és válasszuk úgy a listaelemeket, hogy minden $b_i = 1$, és a_i -k legyenek

$$a_i = \begin{array}{ll} \frac{1}{2} + \epsilon & 1 \leq i \leq 60N \\ \frac{1}{4} + 2\epsilon & 60N < i \leq 120N \\ \frac{1}{4} + \epsilon & 120N < i \leq 180N \\ \frac{1}{4} + 2\epsilon & 180N < i \leq 300N \\ \frac{1}{2} + \epsilon & 300N < i \leq 300N + 6\bar{N} \\ \frac{1}{4} + 2\epsilon & 300N + 6\bar{N} < i \leq 300N + 12\bar{N} \\ \frac{1}{4} + \epsilon & 300N + 12\bar{N} < i \leq 300N + 18\bar{N} \\ \frac{1}{4} - 2\epsilon & 300N + 18\bar{N} < i \leq 300N + 30\bar{N} \\ 1 & 300N + 30\bar{N} < i \leq n \end{array}$$

Alkalmazzuk a GH heurisztikát erre a listára. Ekkor keletkezik

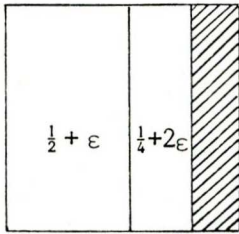
$$\begin{array}{llll}
 60N & + & 6\bar{N} & \text{db} \quad \frac{1}{2} + \epsilon \\
 60N & + & 6\bar{N} & \text{db} \quad \frac{1}{4} + 2\epsilon \\
 60N & + & 6\bar{N} & \text{db} \quad \frac{1}{4} + \epsilon \\
 120N & + & 12\bar{N} & \text{db} \quad \frac{1}{4} - 2\epsilon \\
 & & \bar{k} & \text{db} \quad 1 \text{ szélességű}
 \end{array}$$

Ez azt jelenti, hogy ezt a listát optimális esetben $90N+9\bar{N}+\bar{k}$ csikban tudjuk elhelyezni, míg a GH algoritmussal $110N+11\bar{N}+\bar{k}$ csik szükséges.

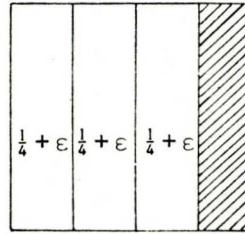
A $60N + 6\bar{N} \quad \frac{1}{2}+\epsilon$ szélességű csikhoz azonban a JOHNSON tétel javítása szerint mindig megadható egy olyan rendezetlen sorozat, amely - bár optimális esetben $60N + 6\bar{N}$ ládában /csikban/ fér el - nem helyezhető el a GH algoritmussal csak $\frac{17}{10}(60N+6\bar{N})-4$ -nél több csikban. Hasonlóan mindig adható olyan lista, amelyre a GH algoritmus a $60N + 6\bar{N}$ darab $\frac{1}{4}+2\epsilon$ szélességű csikot $102N + \frac{102}{10} \bar{N} - 4$ csikban helyezi el. Folytatva a gondolatmenetet, azt kapjuk, hogy lesz olyan lista, amely csak

$$\begin{array}{llll}
 102N & + & \frac{102}{10} \bar{N} - 4 & \frac{1}{2} + \epsilon \\
 102N & + & \frac{102}{10} \bar{N} - 4 & \frac{1}{4} + 2\epsilon \\
 102N & + & \frac{102}{10} \bar{N} - 4 & \frac{1}{4} + \epsilon \\
 204N & + & \frac{204}{10} \bar{N} - 4 & \frac{1}{4} - 2\epsilon \\
 & & \frac{17}{10} \bar{k} - 4 & 1 \text{ szélességű csikban} \\
 & & & \text{helyezhető el}
 \end{array}$$

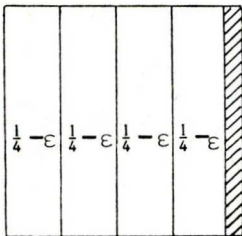
Ezeket a GH algoritmus a következőképpen helyezi el:



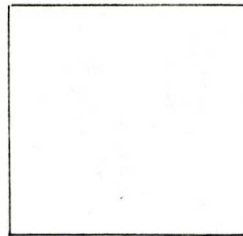
$$102N + \frac{102\overline{N}}{10} - 4 \text{ db}$$



$$\frac{102N + \frac{102\overline{N}}{10} - 4}{3} \text{ db}$$



$$\frac{204N + \frac{204\overline{N}}{10} - 4}{4} \text{ db}$$



$$\frac{17\overline{k}}{10} - 4 \text{ db}$$

Igy

$$\begin{aligned} |\mathcal{L}_{GH}(L)| &= 102N + \frac{102\overline{N}}{10} - 4 + 34N + \frac{34\overline{N}}{10} - \frac{4}{3} + 51N + \frac{51\overline{N}}{10} - 1 + \frac{17\overline{k}}{10} - 4 = \\ &= 187N + \frac{187\overline{N}}{10} + \frac{17\overline{k}}{10} - \frac{31}{3} = \\ &= 90N + 9\overline{N} + \overline{k} + 97N + \frac{97\overline{N}}{10} + \frac{7\overline{k}}{10} - \frac{31}{3} = \\ &= L^* + \frac{97}{90} L^* - \overline{k} - \frac{31}{3} \geq \frac{187}{90} L - 20 \end{aligned}$$

Vegyük észre azt, hogy akárhogyan is adunk meg egy első koordinátájában rendezett L listát, ez mindig elhelyezhető

$L_1 \leq \frac{11}{9} L^* + 4$ ládában. De bárhogyan is adunk meg egy rendezetlen listát az L_1 számú ládára, az legfeljebb $L_2 \leq \frac{17}{10} L_1 + 2$ ládában helyezhető el. Így, ha L egy tetszőleges lista, akkor

$$|\mathcal{L}_{GH}(L)| \leq \frac{17}{10} L_1 + 2 \leq \frac{17}{10} \left(\frac{11}{9} L^* + 4 \right) + 2 = \frac{187}{90} L^* + 9$$

Ezek alapján kimondható a következő

2.2.2. TÉTEL: Akárhogyan is adunk meg egy $L = \{(a_1, b_1), \dots, (a_n, b_n)\}$ listát, ez a GH algoritlussal mindig elhelyezhető legfeljebb $\frac{187}{90} L^* + 9$ ládában, ahol L^* azoknak a ládáknak a száma, amelyekben optimális esetben elhelyezhető az L lista.

Jelöljük

$$R_{GH}(k) = \max_{L^*} \frac{|\mathcal{L}_{GH}(L)|}{L^*}$$

ahol a maximumot azokra az L listákra számítjuk, amelyekre $L^* = k$

Ekkor az előző két tétel értelmében igaz a következő

2.2.3. TÉTEL:

$$\lim_{k \rightarrow \infty} R_{GH}(k) = \frac{187}{90}$$

Bizonyítás: A tétel állítása közvetlenül következik az előző két tételből.



ÖSSZEFOGLALÁS

Dolgozatunk első felében irodalmi áttekintést nyújtottunk a szabási feladat egzakt megoldási módszereiről. Négy megoldást ismertettünk:

- Lineáris programozási modell
- Gilmore-Gomory módszer
- 0-1 programozási modell
- szubgradiens optimalizálás

A fenti módszerek közül kiemelten foglalkoztunk a Gilmore-Gomory módszerrel, és ennek kiterjesztésével. Megvizsgáltuk, hogyan alkalmazható az eljárás, ha

- különböző hosszúságú félkésztermékek állnak rendelkezésre;
- korlátozva van a vágófejek száma;
- a félkésztermékekből csak véges mennyiség áll rendelkezésre;
- a megrendelések kielégítésénél darabszám-tűréseket engedünk meg.

Ismertettünk egy algoritmust, amely kétdimenziós feladatok egzakt megoldására használható.

A dolgozat második részében heurisztikus algoritmusok vizsgálatával foglalkoztunk. Először azokat az eredményeket mutattuk be, amelyeket az egydimenziós heurisztikák általános

elemzésekor kaphatunk, majd négy heurisztikát vizsgáltunk részletesen:

- első illesztés /FF/
- legjobb illesztés /BF/
- első illesztés rendezéssel /FFD/
- legjobb illesztés rendezéssel /BFD/

A dolgozat utolsó része a heurisztikák vizsgálatával kapcsolatos saját eredményeinket tárgyalta:

- Bemutattunk egy módszert, amely az FF algoritmus alsó korlátjának javítását eredményezi;
- Ismertettünk egy kétdimenziós heurisztikus eljárást, amelynek elvégeztük a legjobb eset vizsgálatát, és felső korlátot is adtunk. Elemzésünk végső eredménye egy aszimptotikus konvergenciát kimondó tétel volt, amely a heurisztika által felhasznált ládák számát adta meg.

IRODALOMJEGYZÉK

1. ALBANO, A. és ORSINI, R.:

A Heuristic Solution of the Rectangular Cutting
Stock Problem.

Computer Journal.

2. ART, R.C.:

An Approach to the two-dimensional Irregular
Cutting Stock Problem.

IBM Report, No. 320-2006. 1966.

3. CHRISTOFIDES, N. és WHITLOCK, C.:

An Algorithm for Two-dimensional Cutting Problems.

Opns. Res., 25, 30-44 /1977/

4. DYSON, R.G. és GREGORY, A.S.:

The Cutting Stock Problem in the Flat Glass
Industry.

Opns. Res.Quaterly, Vol.25. No.1., 41-53

5. EILON, S. és CHRISTOFIDES, N.:

The Loading Problem.

Management Science 17, 259-268 /1977/

6. FREDRICKSON, G.N.:

Probabilistic Analysis for Simple One- and Two-
dimensional Bin Packing Algorithm.

Information Proceedings Letters 11, 156-161 /1980/

7. FRIDMAN, F.A.:

Solution of the Knapsack Problem with Special
Supplementary Constrains. Proc. of the VI. Winter
School on Math. and Related Question 244-258

8. GALAMBOS, G., CSIRIK, J., KUBA, A. és LENGYEL, I.:
Üvegszabás optimalizálása lineáris programozási
modell segítségével.
X. Magyar Operációkutatási Konferencia, Debrecen,
1980.
9. GILMORE, P.C. és GOMORY, R.E.:
A Linear Programming Approach to the Cutting Stock
Problem.
Opns. Res., 9, 849-859 /1961/
10. GILMORE, P.C. és GOMORY, R.E.:
A Linear Programming Approach to the Cutting Stock
Problem. Part II.
Opns. Res., 11, 863-888 /1963/
11. GILMORE, P.C. és GOMORY, R.E.:
Multistage Cutting Stock Problems of Two and more
Dimensions.
Opns. Res., 13, 94-120 /1965/
12. GILMORE, P.C. és GOMORY, R.E.:
The Theory and Computation of Knapsack Functions.
Opns. Res., 14, 1045-1074 /1966/
13. GOLDEN, B.L.:
Approaches to the Cutting Stock Problem.
AIIE Transactions 8, 265-274 /1976/
14. GVOZDEV, S.E.:
A Generalization of the Knapsack Problems.
Upravljaemüje Szisztemü 15, 16-31 /1976/
15. HAHN, S.G.:
On the Optimal Cutting of Defective Sheets.
Opns. Res., 16, 1100-1114 /1968/

16. HELD, M. és KARP, R.:
The Travelling Salesman Problem and Minimum
Spanning Trees.
Opns. Res., 18, 1138-1162 /1970/
17. HELD, M., WOLFE, P. és CROWDER, H.:
Validation of Subgradient Optimization.
Mathematical Programming 6, 62-88 /1974/
18. HERZ, J.C.:
Recursive Computational Procedure for Two-dimensional
Stock Cutting.
IBM J.Res.Development 462-469 /1972/
19. INGARGIOLA,
A General Algorithm for One-dimensional Knapsack
Problem.
Opns. Res., 25, 752-759 /1977/
20. JOHNSON, D.S.:
Fast Algorithm for Bin-Packing.
Journal Comp.Syst.Sci., 8, 272-314 /1974/
21. JOHNSON, D.S., DEHMERS,A., UHLMAN,J.D., GAREY,M.R., GRAHAM,R.L.
Worst Case Performance Bounds for Simple One-dimensional
Packing Algorithms.
Cover Sheets for Technical Memorandum Bell Laboratories,
1973
22. KANTOROVICS, Z.V.:
Mathematical Methods of Organizing and Planning
Production.
Management Science 6, 366-422 /1962/

23. KARP, R.M.:

Reducibility among Combinatorial Problems.

In: Complexity of Computer Computation /Eds.:

R.E.Miller and J.W.Thatcher/, Plenum Publishing Co.,

New York, 1972.

24. KÓSA ANDRÁS:

Optimumszámítási modellek.

Műszaki Könyvkiadó, Budapest, 1979.

25. KOU, L.T. és MARKOWSKY, G.:

Multidimensional Bin-Packing Algorithms.

IBM J. Res. Development 443-448 /1977/

26. MADSEN, O.B.G.:

Glass Cutting in a Small Firm.

Mathematical Programming 17, 85-90 /1979/

27. MARTOS BÉLA:

Hiperbolikus Programozás.

MTA Mat.Int.Közleményei 5, 383-406 /1960/

28. Operációkutatás

Szerkesztette: dr. Csath Magdolna

SZÁMOK 1972

29. SHAPIRO, S.D.:

Performance of Heuristic Bin-Packing Algorithms
with Segments of Random Length.

Information and Control 35, 146-158 /1977/

30. SZELEZSÁN JÁNOS:

Lineáris programozás.

Műszaki Könyvkiadó, Budapest, 1965.

31. TENG, J.T.:

A Matrix Algorithm for Larger Knapsack Problems

Bulletin Inst.Math.Acad.Sinica 6, 197-201 /1978/

32. TILANUS, C.B. és GERHARDT, C.:

An Application of Cutting Stock in the Steel
Industry.

Operational Res., '75 /ed. by K.B. Halley/,

North Holland P.C., 1976.

FÜGGELÉK

A szabási feladat gyakorlati alkalmazása

A szabási feladattal egy gyakorlati probléma kapcsán kezdtünk el foglalkozni. Az Orosházi Siküveggyárban félkész üvegtáblákból kétütemű guillotine vágások segítségével megrendeléseket szabnak le. A feladat az volt, hogy készítsünk optimális szabástervet. A megoldást először lineáris programozási modell segítségével kerestük. Az LP feladatot a programrendszerben egy olyan programmal oldottuk meg, amely egy programcsomag /OPSI/ része. A megoldást azonban csak olyan esetekre tudtuk szolgáltatni, amikor a különböző szabások száma 1400 alatt volt. A technológiai korlátok jelentősen csökkentették az összes lehetséges szabásoknak a számát, azonban az elvárt korlátot csak úgy tudtuk elérni, hogy a megrendelések számát korlátoztuk /maximum 20 megrendelést engedtünk meg/, és a keletkezett szabáshalmazt is redukáltuk [8]. Így nagyobb megrendelésszám esetén a megrendelésállományt csoportokra kellett bontani, ami természetesen az optimális megoldásban ronthatta a cél-függvényértéket.

Az elkészített programrendszert az Orosházi Siküveggyárban kísérleti futtatásokkal kipróbáltuk. A gyakorlati

tapasztalatok az alkalmazott -számítástechnikai - korlátok ellenére is igen kedvezőek voltak. A korábbi "ad hoc" jellegű előkészítés 40% körüli selejttjével szemben a számítógépes feldolgozás 20-30% körüli selejtet adott a megoldásban.

A programrendszer legsebezhetőbb pontjának a megrendelések darabszámának korlátozásából adódó feltétel tűnt. Azért, hogy ezt kiküszöböljük, két irányban kerestünk megoldást. Az egyik javítás a Gilmore-Gomory módszer. Azért, hogy a módszerrel kellő mélységben megismerjük, először egydimenziós esetre kerestük a megoldást különböző korlátozó feltételek mellett. Jelenlegi eredményeink azt mutatják, hogy nagyobb megrendelésszám esetén /40-50/ az optimum kiszámításához szükséges idő jelentősen megnövekszik. /A CPU idő 20 perc körül mozog./ Ez azonban pótolhatja azt a jelentős költség-növekedést, amely a megrendelésállomány részekre bontásából adódhat, abból, hogy a selejtszázalék megnövekszik.

A heurisztikus algoritmusok alkalmazása a másik út arra, hogy az összes megrendelést együtt vizsgálhassuk a programozás során. Az ismertetett kétdimenziós algoritmus lényegében megfelel azoknak a követelményeknek, amelyeket a technológiai feltételek állítanak egy félkész tábla leszábásával szemben az Orosházi Siküveggyárban. Ezzel az volt a célunk, hogy megvizsgáljuk annak a lehetőségét, hogy egy gyors, könnyen kezelhető, többször futtatható program milyen lehetőségeket biztosítana.

A heurisztikus program és a Gilmore-Gomory módszer gyakorlati felhasználására nem került sor.